Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 196 (2015)

Polynomials, Quantum Query Complexity, and Grothendieck's Inequality

Scott Aaronson[1] Andris Ambainis[2] Jānis Iraids[2] Mārtiņš Kokainis[2]
Juris Smotrovs[2]

Abstract

We show an equivalence between 1-query quantum algorithms and representations by degree-2 polynomials. Namely, a partial Boolean function f is computable by a 1-query quantum algorithm with error bounded by $\epsilon < 1/2$ iff f can be approximated by a degree-2 polynomial with error bounded by $\epsilon' < 1/2$. This result holds for two different notions of approximation by a polynomial: the standard definition of Nisan and Szegedy [20] and the approximation by block-multilinear polynomials recently introduced by Aaronson and Ambainis [1].

We also show two results for polynomials of higher degree. First, there is a total Boolean function which requires $\tilde{\Omega}(n)$ quantum queries but can be represented by a block-multilinear polynomial of degree $\tilde{O}(\sqrt{n})$. Thus, in the general case (for an arbitrary number of queries), block-multilinear polynomials are not equivalent to quantum algorithms.

Second, for any constant degree k, the two notions of approximation by a polynomial (the standard and the block-multilinear) are equivalent. As a consequence, we solve an open problem from [1], showing that one can estimate the value of any bounded degree-k polynomial $p : \{0,1\}^n \to [-1,1]$ with $O(n^{1-\frac{1}{2k}})$ queries.

[1]Computer Science and Artificial Intelligence Laboratory, MIT. Supported by an Alan T. Waterman Award from the National Science Foundation, under grant no. 1249349. E-mail: aaronson@csail.mit.edu.

[2]Faculty of Computing, University of Latvia. Supported by the European Commission FET-Proactive project QALGO, ERC Advanced Grant MQC and Latvian State Research programme NexIT project No.1. Emails:ambainis@lu.lv, janis.iraids@gmail.com,juris.smotrovs@sets.lv, martins.kokainis@gmail.com.

ISSN 1433-8092

1 Introduction

Many of the known quantum algorithms can be studied in the query model where one measures the complexity of an algorithm by the number of queries to the input that it makes. In particular, this model encompasses Grover's search [16], the quantum part of Shor's factoring algorithm (period-finding) [23], their generalizations and many of the more recent quantum algorithms such as element distinctness [6] and NAND tree evaluation [14, 7, 22].

For proving lower bounds on quantum query algorithms, one often uses a connection to polynomials [8]. After k queries to an input $x_1, \ldots, x_N$, the amplitudes of the algorithm's quantum state are polynomials of degree at most k in $x_1, \ldots, x_N$. Therefore, one can prove that there is no quantum algorithm using fewer than k queries by showing the non-existence of a polynomial with certain properties.

For example, one can use this approach to show that any quantum algorithm for Grover's search algorithm requires $\Omega(\sqrt{N})$ queries [8] or to show an optimal quantum lower bound for finding collisions [3]. In some cases, the lower bounds obtained by polynomials method are tight, either exactly (for example, for computing the parity of N input bits $x_1, \ldots, x_N$ [8]) or up to a constant factor (Grover's search and many other examples). In other cases, the number of queries to compute a function $f(x_1, \ldots, x_N)$ is asymptotically larger than the lower bound which follows from polynomials [5, 2].

In this paper, we discover the first case where we can go in the opposite direction: from a polynomial to a bounded-error quantum algorithm[1]. That is, polynomials with certain properties and quantum algorithms are equivalent!

In more detail, we consider computing partial Boolean functions $f(x_1, \ldots, x_n)$ and show that the existence of a quantum algorithm that computes f with 1 query is equivalent to the existence of a degree 2 polynomial that approximates f. This result holds for two different notions of approximation by a polynomial: the standard one in [20] and the approximation by block-multilinear polynomials introduced in [1].

The methods that we use are quite interesting. To transform a polynomial into a quantum algorithm, we first transform it into the block-multilinear form of [1] and then use a variant of Grothendieck's inequality for relating two matrix norms [21]. One of the two norms corresponds to the constraints on the block-multilinear polynomials while the other norm corresponds to algorithm's transformations being unitary. While Grothendieck's inequality has been used in the context of quantum non-locality (e.g. in [4]), this appears to be its first use in the context of quantum algorithms.

We then show two results for polynomials of larger degree:

- similarly to general polynomials, block-multilinear polynomials are not equivalent to quantum algorithms in the general case: one of *cheat-sheet* functions of [2] requires $\tilde{\Omega}(n)$ quantum queries but can be described by a block-multilinear polynomial of degree $\tilde{O}(\sqrt{n})$;
- for representations by polynomials of degree $d = O(1)$, a partial function f can be represented by a general polynomial of degree d if and only if it can be represented by a block-multilinear polynomial of degree d.

[1]In unbounded-error settings, equivalences between quantum algorithms and polynomials were previously shown by de Wolf [25] and by Montanaro et al. [19].

We note that the first result does not exclude an equivalence between quantum algorithms and polynomials for a small number of queries that is larger than 1. For example, 2-query quantum algorithms could be equivalent to polynomials of degree 4. The second result shows that, to prove such an equivalence, it suffices to give a transformation from block-multilinear polynomials to quantum algorithms.

Another consequence of the second result is that, if we have a general polynomial $f(x_1, \ldots, x_n)$ which is bounded (i.e., $|f| \leq 1$ for all $x_1, \ldots, x_n \in \{0,1\}$), the value of this polynomial can be estimated with $O(n^{1-1/2d})$ queries about values of $x_1, \ldots, x_n$. This resolves an open problem from [1] and is shown by transforming f into a block-multilinear form and then using the sampling algorithm of [1] for block-multilinear polynomials.

2 Preliminaries

2.1 Notation

By $[a \,..\, b]$, with a, b being integers, $a \leq b$, we denote the set $\{a, a+1, a+2, \ldots, b\}$. When $a = 1$, notation $[a \,..\, b]$ is simplified to $[b]$.

For a vector x, let $\|x\|_p$ stand for the p-norm; when $p = 2$, this is the Euclidean norm and the notation is simplified to $\|x\|$. For a matrix A, by $\|A\|_{p\to q}$ we denote

$$\|A\|_{p\to q} = \sup_{x:\|x\|_p\neq 0} \frac{\|Ax\|_q}{\|x\|_p} = \max_{x:\|x\|_p=1} \|Ax\|_q = \max_{x:\|x\|_p\leq 1} \|Ax\|_q .$$

By $\|A\|$ we understand the usual operator norm $\|A\|_{2\to 2}$.

D_x stands for the diagonal matrix with components of x on its diagonal.

By K we denote the (real) Grothendieck's constant which is defined as the smallest number with the following property: if $A = (a_{ij})$ is such that $\sum_{i,j} a_{ij} x_i y_j \leq 1$ for any choice of $x_i, y_j \in \{-1, 1\}$, then $\sum_{i,j} a_{ij} (u_i, v_j) \leq K$ for any choice of vectors (with real components) u_i, v_j with $\|u_i\| = 1$ and $\|v_j\| = 1$ for all i, j. It is known [21, 10] that

$$\frac{\pi}{2} \leq K < \frac{\pi}{2\ln(1+\sqrt{2})}.$$

2.2 Quantum query complexity and polynomial degree

We consider computing partial Boolean functions $f(x_1, \ldots, x_n) : X \to \{0, 1\}$ (for some $X \subseteq \{0,1\}^n$) in the standard quantum query model. For technical convenience, we relabel the values of input variables x_i from $\{0,1\}$ to $\{-1,1\}$. Then, a partial Boolean function f maps a set $X \subseteq \{-1,1\}^n$ to $\{0,1\}$.

Let $Q_\epsilon(f)$ be the minimum number of queries in a quantum algorithm computing f correctly with probability at least $1-\epsilon$, for every $x = (x_1, \ldots, x_n)$ for which $f(x)$ is defined.

Definition 1. $\widetilde{\deg}_\epsilon(f)$ is the minimum degree of a polynomial p (in variables $x_1, \ldots, x_n$) such that

1. $|p(x) - f(x)| \leq \epsilon$ for all $x \in \{-1,1\}^n$ for which $f(x)$ is defined;
2. $p(x) \in [0,1]$ for all $x \in \{-1,1\}^n$.

$\deg(f)$ denotes $\widetilde{\deg}_0(f)$.

It is well known that $\mathrm{Q}_\epsilon(f) \geq \frac{1}{2}\widetilde{\deg}_\epsilon(f)$ [8]. We now consider a refinement of this result due to [1]. We say that a polynomial p of degree k is *block-multilinear* if its variables $x_1, \ldots, x_N$ can be partitioned into k blocks, $R_1, \ldots, R_k$, so that every monomial of p contains exactly one variable from each block.

Lemma 2 ([1, Lemma 20])**.** *Let $\mathcal{A}$ be a quantum algorithm that makes t queries to a Boolean input $x \in \{-1,1\}^n$. Then there exists a degree-$2t$ block-multilinear polynomial $p : \mathbb{R}^{2t(n+1)} \to \mathbb{R}$, with $2t$ blocks of $n+1$ variables each, such that*

(i) the probability that $\mathcal{A}$ outputs 1 for an input $x = (x_1, \ldots, x_n) \in \{-1,1\}^n$ equals $p(\tilde{x}, \ldots, \tilde{x})$, where $\tilde{x} := (1, x_1, \ldots, x_n)$ (with $\tilde{x}$ repeated $2t$ times), and

(ii) $p(z) \in [-1,1]$ for all $z \in \{-1,1\}^{2t(n+1)}$.

The first variable in each block (which is set to 1 in the requirement (i)) corresponds to the possibility that the algorithm is not asking any of the actual variables $x_1, \ldots, x_n$ in a given query. (Although the statement of Lemma 20 in [1] does not mention such variables explicitly, they are used in the proof of the Lemma.)

Definition 3. Let the *block-multilinear approximate degree* of f, or $\widetilde{\mathrm{bmdeg}}_\epsilon(f)$, be the minimum degree of any block-multilinear polynomial $p : \mathbb{R}^{k(n+1)} \to \mathbb{R}$, with k blocks of $n+1$ variables each, such that

(i) $|p(\tilde{x}, \ldots, \tilde{x}) - f(x)| \leq \epsilon$ for all $x \in \{-1,1\}^n$ for which $f(x)$ is defined, and

(ii) $p(x_{1,0}, x_{1,1}, \ldots, x_{1,n}, x_{2,0}, \ldots, x_{k,n}) \in [-1,1]$ for all $x_{1,0}, \ldots, x_{k,n} \in \{-1,1\}^{k(n+1)}$.

$\mathrm{bmdeg}(f)$ denotes $\widetilde{\mathrm{bmdeg}}_0(f)$.

As a particular case, this definition includes block-multilinear polynomials $p : \mathbb{R}^{kn} \to \mathbb{R}$ which satisfy

$$\forall x \in \{-1,1\}^n \ \ |p(x, \ldots, x) - f(x)| \leq \epsilon \quad \text{and} \quad \forall z \in \{-1,1\}^{kn}\, p(z) \in [-1,1],$$

because we can view them as polynomials $p : \mathbb{R}^{k(n+1)} \to \mathbb{R}$ in which each monomial containing a variable $x_{1,0}$, $x_{2,0}$, $\ldots$, or $x_{k,0}$ has a coefficient zero.

We have $\widetilde{\deg}_\epsilon(f) \leq \widetilde{\mathrm{bmdeg}}_\epsilon(f) \leq 2\,\mathrm{Q}_\epsilon(f)$. The first of the two inequalities follows by taking $q(x) = p(\tilde{x}, \ldots, \tilde{x})$. If p satisfies the requirements of Definition 3, then q satisfies the requirements of Definition 1. The second inequality follows from Lemma 2.

2.3 Block-multilinear polynomials of degree 2

Let

$$p(x_1, \ldots, x_n, y_1, \ldots, y_m) = \sum_{\substack{i \in [n] \\ j \in [m]}} a_{ij} x_i y_j, \tag{1}$$

be a block-multilinear polynomial of degree 2, with the variables in the first block labeled as $x_1, \ldots, x_n$ and the variables in the second block labeled as $y_1, \ldots, y_m$. We say that p is *bounded* if $|p(x_1, \ldots, x_n, y_1, \ldots, y_m)| \leq 1$ for all $x_1, \ldots, y_m \in \{-1, 1\}$. Then, we have

$$\max_{\substack{x\in\{-1,1\}^n \\ y\in\{-1,1\}^m}} \left| \sum_{\substack{i\in[n] \\ j\in[m]}} a_{ij} x_i y_j \right| \leq 1.$$

Let A be the $n \times m$ matrix with entries a_{ij}, then

$$p(x, y) = x^T A y \quad \text{for all } x \in \mathbb{R}^n,\ y \in \mathbb{R}^m$$

and p being bounded translates to the infinity-to-1 norm of A being at most 1, i.e., $\|A\|_{\infty\to1} \leq 1$.

3 Equivalence between polynomials of degree 2 and 1-query quantum algorithms

Let f be a partial Boolean function. In this section, we show that the following three statements are essentially equivalent[2]:

(a) $\mathrm{Q}_\epsilon(f) \leq 1$ for some ϵ with $0 \leq \epsilon < \frac{1}{2}$;

(b) $\widetilde{\mathrm{bmdeg}}_{\epsilon'}(f) \leq 2$ for some ϵ' with $0 \leq \epsilon' < \frac{1}{2}$;

(c) $\widetilde{\deg}_{\epsilon''}(f) \leq 2$ for some ϵ'' with $0 \leq \epsilon'' < \frac{1}{2}$.

Given (a), Lemma 2 implies that (b) and (c) hold with $\epsilon'' = \epsilon' = \epsilon$. We now show that (b) implies (a) with $\epsilon = \frac{K+\epsilon'}{2(K+1)}$ where K is Grothendieck's constant. After that, we show that (c) implies (b) with $\epsilon' = \frac{1+\epsilon''}{3}$.

Theorem 4. *Let f be a partial Boolean function. If $\widetilde{\mathrm{bmdeg}}_{\epsilon'}(f) \leq 2$, then $\mathrm{Q}_\epsilon(f) \leq 1$ for $\epsilon = \frac{K+\epsilon'}{2(K+1)}$.*

Proof. We start with two technical lemmas.

Lemma 5. *If a $n \times m$ complex matrix B satisfies $\|B\| \leq C$, then there exists a unitary U (on a possibly larger space with basis states $|1\rangle, \ldots, |k\rangle$ for some $k \geq \max(n, m)$) such that, for any unit vector $|y\rangle = \sum_{i=1}^m \alpha_i |i\rangle$, $U|y\rangle = \frac{B|y\rangle}{C} + |\phi\rangle$, with $|\phi\rangle$ consisting of basis states $|i\rangle$, $i > n$ only.*

Proof. Without the loss of generality, we can assume that $C = 1$ (otherwise, we just replace the matrix B by $\frac{B}{C}$).

Let $A = I - B^\dagger B$. Since $\|B\| \leq 1$, the eigenvalues of $B^\dagger B$ are at most 1 and, hence, A is positive semidefinite. Let $A = V^\dagger \Lambda V$ be the eigendecomposition of A, with V being a unitary matrix and Λ a diagonal matrix. We take $W = \sqrt{\Lambda} V$. Then, $A = W^\dagger W$ and, if we take the block matrix $U = \begin{pmatrix} B \\ W \end{pmatrix}$, we get $U^\dagger U = B^\dagger B + W^\dagger W = I$.

[2]The equivalence here involves some loss in the error ϵ. probability of $\mathcal{A}$. However, the bound ϵ on the error probability of the resulting quantum algorithm only depends on the error of the polynomial approximation from which we started and does not increase with the number of variables n.

Let $k \times m$ be the size of the matrix U. For any $i \in \{1, \ldots, m\}$, we have $\langle i| U^\dagger U |i\rangle = \langle i| I |i\rangle = 1$ and for any $i, j \in \{1, \ldots, m\} : i \neq j$, we have $\langle i| U^\dagger U |j\rangle = \langle i| I |j\rangle = 0$. Therefore, $U|1\rangle, \ldots, U|m\rangle$ are orthogonal vectors of length 1 and we can complete U to a $k \times k$ unitary matrix by choosing $U|m+1\rangle, \ldots, U|k\rangle$ so that they are orthogonal (both one to another and to $U|1\rangle, \ldots, U|m\rangle$) and of length 1. □

Lemma 6. *Let $A = (a_{ij})_{i\in[n], j\in[m]}$ with $\sqrt{nm}\|A\| \leq C$ and let*

$$p(x_1, \ldots, x_n, y_1, \ldots, y_m) = \sum_{i=1}^{n} \sum_{j=1}^{m} a_{ij} x_i y_j.$$

Then, there is a quantum algorithm that makes 1 query to $x_1, \ldots, x_n$, $y_1, \ldots, y_m$ and outputs 1 with probability

$$r = \frac{1}{2}\left(1 + \frac{p(x_1, \ldots, x_n, y_1, \ldots, y_m)}{C}\right).$$

Proof. Let $B = \sqrt{nm}\, A$, $A = (a_{ij})$. Then,

$$\|B\| = \|A\| \sqrt{nm} \leq C.$$

The 1-query quantum algorithm uses a version of the well-known SWAP test [11] for estimating the inner product $|\langle \psi | \psi' \rangle|$ of two quantum states $|\psi\rangle$ and $|\psi'\rangle$. Our test works by preparing the state

$$\frac{1}{\sqrt{2}} |0\rangle |\psi\rangle + \frac{1}{\sqrt{2}} |1\rangle |\psi'\rangle \qquad (2)$$

and then performing the Hadamard transformation on the first qubit and measuring the first qubit[3]. The probability that the result of the measurement is 0 is equal to

$$r_0 = \frac{1}{2}(1 + \Re\langle \psi | \psi' \rangle)$$

where $\Re x$ denotes the real part of a complex number x.

By Lemma 5, there is a unitary U s.t. for any unit vector $|y\rangle = \sum_{i=1}^{m} \alpha_i |i\rangle$ we have $U|y\rangle = \frac{B|y\rangle}{C} + |\phi\rangle$, with $\langle i | \phi\rangle = 0$ for all $i \in [n]$.

The algorithm applies SWAP test to $|x\rangle = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} x_i |i\rangle$ and $U|y\rangle$, $|y\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} y_i |i\rangle$. Each of those states can be prepared with one query (to x_i's or y_i's). Hence, we can also prepare the state (2) with one query. The inner product $\langle \psi | \psi' \rangle$ that is being estimated is equal to

$$\langle x| U |y\rangle = \frac{1}{C}\langle x| B |y\rangle = \frac{1}{C} p(x_1, \ldots, x_n, y_1, \ldots, y_m).$$

□

[3] This test is slightly different from the standard SWAP test in which one prepares both $|\psi\rangle$ and $|\psi'\rangle$ and then performs a SWAP gate conditioned by a qubit that is initially in the $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ state. Because of this difference, we can perform the SWAP test with just 1 query instead of 2 (one for $|\psi\rangle$ and one for $|\psi'\rangle$). Another result of this difference is that the probability of measuring 0 changes from $\frac{1}{2}(1 + |\langle \psi | \psi' \rangle|^2)$ for the standard SWAP test to $\frac{1}{2}(1 + \Re\langle \psi | \psi' \rangle)$ for our test.

Let $p(x_1, \ldots, x_n, y_1, \ldots, y_m) = \sum_{i=1}^n \sum_{j=1}^m a_{ij} x_i y_j$ be the polynomial from Definition 3 which shows that $\widetilde{\mathrm{bmdeg}}_{\epsilon'}(f) = 2$. Then, as we argued in subsection 2.3, the matrix $A = (a_{ij})$ satisfies $\|A\|_{\infty \to 1} \leq 1$. Although this does not imply that $\|A\|$ is sufficiently small, we can preprocess the polynomial p so that we achieve $\sqrt{n'm'}\|A'\| \leq K$ for the n'-by-m' matrix A' of coefficients of the polynomial after the preprocessing.

To preprocess the polynomial, we perform an operation called *variable-splitting* [1]. The operation consists of taking a variable x_j (or y_j) and replacing it by m variables, in the following way. We introduce m new variables $x_{l_1}, \ldots, x_{l_m}$, and define p' as the polynomial obtained by substituting $\frac{x_{l_1}+\cdots+x_{l_m}}{m}$ in the polynomial p instead of x_j. If we substitute $x_{l_1} = \ldots = x_{l_m} = x_j$, p' is equal to $p(x_1, \ldots, x_n, y_1, \ldots, y_m)$. Thus, being able to evaluate p' implies being able to evaluate p (in the same sense of the word "evaluate").

In appendix A, we show

Lemma 7. *If a polynomial*

$$p(x_1, \ldots, x_n, y_1, \ldots, y_m) = \sum_{i=1}^n \sum_{j=1}^m a_{ij} x_i y_j.$$

satisfies $p(x, y) \in [-1, 1]$ for all $x \in \{-1, 1\}^n$, $y \in \{-1, 1\}^m$, then for every $\delta > 0$ there exists a sequence of row and column splittings that transforms $A = (a_{ij})$ to an $n' \times m'$ matrix $A' = (a'_{ij})$ that satisfies

$$\frac{\|A'\| \sqrt{n'm'}}{\|A'\|_{\infty \to 1}} \leq K + \delta.$$

Then, we can apply Lemma 6 with $C = K + \delta$ to evaluate the polynomial

$$p'(x'_1, \ldots, x'_{n'}, y'_1, \ldots, y'_{m'}) = \sum_{i=1}^{n'} \sum_{j=1}^{m'} a_{ij} x'_i y'_j.$$

for $(x'_1, \ldots, x'_{n'}, y'_1, \ldots, y'_{m'})$ which corresponds to the point $(x_1, \ldots, x_n, y_1, \ldots, y_m)$ at which we want to evaluate the original polynomial $p(x_1, \ldots, y_m)$.

If $p(x, y) \in [0, \epsilon']$, then Lemma 6 gives $r \leq (1+\frac{\epsilon'}{K})/2$. If $p(x, y) \in [1-\epsilon', 1]$, then $r \geq (1+\frac{1-\epsilon'}{K})/2$.

We now consider an algorithm which outputs 0 with probability $\frac{1}{2K+1}$ and runs the algorithm of Lemma 6 otherwise (with probability $\frac{2K}{2K+1}$). Let q be the probability of this algorithm outputting 1. If $p(x, y) \in [0, \epsilon']$, then $q = \frac{2K}{2K+1} r \leq \frac{K+\epsilon'}{2K+1}$. If $p(x, y) \in [1-\epsilon', 1]$, then $q = \frac{2K}{2K+1} r \geq \frac{K+1-\epsilon'}{2K+1}$. Thus, we have a quantum algorithm with a probability of error which is at most $\epsilon = \frac{K+\epsilon'}{2K+1}$. □

Theorem 8. *Let f be a partial Boolean function. If $\widetilde{\deg}_{\epsilon''}(f) \leq 2$, then $\widetilde{\mathrm{bmdeg}}_{\epsilon'}(f) \leq 1$ for $\epsilon' = \frac{1+\epsilon''}{3}$.*

Proof. We first show a corresponding result for polynomials p with values in $[-1, 1]$ (instead of polynomials p with values in $[0, 1]$ as in Definition 1).

Lemma 9. *Suppose that $f : \{-1, 1\}^n \to \mathbb{R}$ is a multilinear polynomial of degree 2, satisfying $\max_{x \in \{-1,1\}^n} |f(x)| \leq 1$. Then there exists a block-multilinear polynomial $g : \{-1, 1\}^{2n+2} \to \mathbb{R}$ of* *degree 2 with* $\max_{x\in\{-1,1\}^{2n+2}}|g(x)| \le 1$ *such that for every* $x \in \{-1,1\}^n$ *the following equality holds:*

$$g(1,x_1,\dots,x_n,\dots 1,x_1,\dots,x_n) = \frac{1}{3} f(x_1,\dots,x_n).$$

Proof. Suppose that $f : \{-1,1\}^n \to \{-1,1\}$ is a multilinear polynomial of degree 2. Then it can be represented as

$$f(x) = f_\emptyset + \sum_{i=1}^n f_{\{i\}} x_i + \sum_{i<j} f_{\{i,j\}} x_i x_j, \qquad x \in \mathbb{R}^n.$$

Moreover, the constraint $\max_{x\in\{-1,1\}}|f(x)| \le 1$ implies that $|f(x)| \le 1$ for all $x \in [-1,1]^n$.

Define a block-multilinear polynomial

$$g(z,t) = \sum_{i=0}^n \sum_{j=0}^n g_{ij} z_i t_j, \quad z,t \in [-1,1]^{n+1},$$

where

$$\begin{aligned} g_{00} &= \frac{1}{3} f_\emptyset, \\ g_{i0} = g_{0i} &= \frac{1}{6} f_{\{i\}}, \quad i \in [n] \\ g_{ij} = g_{ji} &= \frac{1}{6} f_{\{i,j\}}, \quad i<j. \end{aligned}$$

Then

$$g(z,t) = \frac{1}{3}\left(f_\emptyset z_0 t_0 + \sum_{i=1}^n f_{\{i\}} \frac{z_i t_0 + z_0 t_i}{2} + \sum_{i<j} f_{\{i,j\}} \frac{z_i t_j + z_j t_i}{2} \right), \quad z,t \in \{-1,1\}^{n+1}.$$

Clearly,

$$g\left((1,x),(1,x)\right) = \frac{1}{3}\left(f_\emptyset + \sum_{i=1}^n f_{\{i\}} x_i + \sum_{i<j} f_{\{i,j\}} x_i x_j \right) = \frac{1}{3} f(x), \quad x \in \{-1,1\}^n.$$

Let $x_i = z_0 z_i$, $y_i = t_0 t_i$. Then (by multiplying $g(z,t)$ with $z_0 t_0$) we see that

$$|g(z,t)| = |F(x,y)|,$$

where

$$F(x,y) = \frac{1}{3}\left(f_\emptyset + \sum_{i=1}^n f_{\{i\}} \frac{x_i + y_i}{2} + \sum_{i<j} f_{\{i,j\}} \frac{x_i y_j + x_j y_i}{2} \right), \quad x,y \in \{-1,1\}^n.$$

Moreover, $F(x,x) = \frac{1}{3} f(x)$ for all $x \in \{-1,1\}^n$.

Notice that the following identity holds:

$$F(x,y) = \frac{4F\left(\frac{x+y}{2}, \frac{x+y}{2}\right) - F(x,x) - F(y,y)}{2}, \quad x,y \in \{-1,1\}^n.$$

It follows that

$$|F(x,y)| = \frac{1}{6}\left|4f\left(\frac{x+y}{2}\right) - f(x) - f(y)\right|,$$

for all $x, y \in \{-1,1\}^n$. Since $\frac{x+y}{2} \in [-1,1]^n$, then

$$\left|4f\left(\frac{x+y}{2}\right) - f(x) - f(y)\right| \le 4\left|f\left(\frac{x+y}{2}\right)\right| + |f(x)| + |f(y)| \le 6,$$

thus $|F(x,y)| \le 1$. It follows that g is bounded. □

By rescaling both the initial and the final polynomial to take the values in $[0,1]$, we obtain

Corollary 10. *Suppose that $f : \{-1,1\}^n \to \mathbb{R}$ is a multilinear polynomial of degree 2 satisfying $f(x) \in [0,1]$ for all $x \in \{-1,1\}^n$. Then there exists a block-multilinear polynomial $g : \{-1,1\}^{2n+2} \to \mathbb{R}$ of degree 2 such that for every $x \in \{-1,1\}^n$ the following equality holds:*

$$g(1, x_1, \ldots, x_n, 1, x_1, \ldots, x_n) = \frac{1}{3} + \frac{1}{3}f(x_1, \ldots, x_n).$$

If $f(x_1, \ldots, x_n) \in [0, \epsilon'']$, then $g(1, x_1, \ldots, x_n, 1, x_1, \ldots, x_n) \in [\frac{1}{3}, \frac{1+\epsilon''}{3}]$. If $f(x_1, \ldots, x_n) \in [1-\epsilon'', 1]$, then $g(1, x_1, \ldots, x_n, 1, x_1, \ldots, x_n) \in [\frac{2-\epsilon''}{3}, \frac{2}{3}]$. Thus, we have $\mathrm{bmdeg}_{\epsilon'}(f) = 1$ with $\epsilon' = \frac{1+\epsilon''}{3}$. □

4 Results on polynomials of higher degrees

4.1 Equivalence between general and block-multilinear polynomials

We can extend our result on transforming a bounded polynomial $f(x_1, \ldots, x_n)$ to a bounded block-multilinear polynomial to polynomials of higher degree.

Lemma 11. *Suppose that $g : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial of degree d s.t. $|g(x)| \le 1$ for all $x \in \{-1,1\}^n$.*

Then there exists a bounded block-multilinear polynomial $h : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ and a number $B(d)$ s.t. for all $x_1, \ldots, x_n \in \mathbb{R}$ the following equality holds:

$$h\left(1, x_1, \ldots, x_n, 1, x_1, \ldots, x_n, \ldots, 1, x_1, \ldots, x_n\right) = \frac{1}{B(d)}g(x_1, \ldots, x_n).$$

Moreover, $B(d)$ satisfies

$$B(d) = \Theta\left(\frac{\alpha^d}{\sqrt{d}}\right),$$

where $\alpha = \frac{1}{W(\exp(-1))} \approx 3.5911$ and W stands for the Lambert W function.

Proof. In appendix B. □

Thus, approximations by general polynomials and approximations by block-multilinear polynomials are equivalent for degree $d = O(1)$, up to some loss in the approximation error:

Corollary 12. *Let f be a partial Boolean function with $\widetilde{\deg}_\epsilon(f) \le d$ for some ϵ with $0 \le \epsilon < \frac{1}{2}$. Then, $\widetilde{\mathrm{bmdeg}}_{\epsilon'}(f) \le d$ for $\epsilon' = \frac{1}{2} - \frac{1}{4B(d)} + \frac{\epsilon}{2B(d)}$.*

Proof. Let $p(x)$ be the polynomial that represents $f(x)$, in the sense of Definition 1. We take $g(x) = p(x) - \frac{1}{2}$, apply Lemma 11 and then take $h'(x) = \frac{1}{2} + \frac{h(x)}{2}$ where h is the polynomial produced by Lemma 11.

If $p(x) \in [0, \epsilon]$, then $g(x) \in [-\frac{1}{2}, -\frac{1}{2}+\epsilon]$ and $h(1, x, \ldots, 1, x) \in [-\frac{1}{2B(d)}, -\frac{1}{2B(d)}+\frac{\epsilon}{B(d)}]$. Therefore,

$$h'(1, x, \ldots, 1, x) \in \left[\frac{1}{2} - \frac{1}{4B(d)}, \frac{1}{2} - \frac{1}{4B(d)} + \frac{\epsilon}{2B(d)}\right].$$

Similarly, if $p(x) \in [1-\epsilon, 1]$, then

$$h'(1, x, \ldots, 1, x) \in \left[\frac{1}{2} + \frac{1}{4B(d)} - \frac{\epsilon}{2B(d)}, \frac{1}{2} + \frac{1}{4B(d)}\right].$$

Also, if $|h(y)| \le 1$ for any $y \in \{-1, 1\}^{d(n+1)}$, then $|h'(y)| \le 1$, as well. Therefore, h' represents f in the sense of Definition 3 with $\epsilon' = \frac{1}{2} - \frac{1}{4B(d)} + \frac{\epsilon}{2B(d)}$. □

Lemma 11 and Corollary 12 have two consequences. First, to extend the equivalence between quantum algorithms and polynomials to larger $d = O(1)$, it suffices to show how to transform block-multilinear polynomials into quantum algorithms.

Second, Aaronson and Ambainis [1] showed that a quantum algorithm which makes d queries can be simulated by a classical algorithm making $O(n^{1-1/2d})$ queries, based on the following result

Theorem 13. *[1] Let $h : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ be a block-multilinear polynomial of degree d with $|h(y)| \le 1$ for any $y \in \{-1, 1\}^{d(n+1)}$. Then, $h(y)$ can be approximated within precision $\pm\epsilon$ with high probability, by querying $O((\frac{n}{\epsilon^2})^{1-1/d})$ variables (with a big-O constant that is allowed to depend on d).*

It has been open whether a similar theorem holds for general (not block-multilinear) polynomials $h(x_1, \ldots, x_n)$. Aaronson and Ambainis [1] showed that this is true for degree 2 (using quite sophisticated tools from Fourier analysis) but left it as an open problem for higher degrees. With Lemma 11, we can immediately resolve this problem.

Corollary 14. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a polynomial of degree d with $|g(y)| \le 1$ for any $y \in \{-1, 1\}^{d(n+1)}$. Then, $g(y)$ can be approximated within precision $\pm\epsilon$ with high probability, by querying $O((\frac{n}{\epsilon^2})^{1-1/d})$ variables (with a big-O constant that is allowed to depend on d).*

Proof. We apply Lemma 11 to construct a corresponding block-multilinear polynomial h and then use Theorem 13 to estimate h with precision $\frac{\epsilon}{B(d)}$. Since $B(d)$ is a constant for any fixed d, we can absorb it into the big-O constant. □

4.2 bmdeg and deg vs. Q

The biggest known separation between deg and Q is $Q(f) = \tilde{\Omega}(\deg^2(f))$, recently shown by Aaronson et al. [2] using a novel *cheat-sheet* technique. We extend this result to

Theorem 15. *There exists f with $Q(f) = \tilde{\Omega}(\mathrm{bmdeg}^2(f))$.*

Proof. In appendix C. □

Aaronson et al. [2] also show a separation $Q(f) = \tilde{\Omega}(\widetilde{\deg}(f))^4)$ which does not seem to give $Q(f) = \tilde{\Omega}(\widetilde{\mathrm{bmdeg}}(f))^4)$. (For the natural way of transforming the approximating polynomial of [2] into a block-multilinear form, the resulting block-multilinear polynomial $p(z^{(1)}, z^{(2)}, \ldots)$ can take values that are exponentially large (in its degree) if the blocks $z^{(1)}, z^{(2)}, \ldots$ are not all equal.)

Because of Theorem 15, there is no transformation from a polynomial of degree $2k$ that approximates $f(x_1, \ldots, x_n)$ with error $\epsilon < 1/2$ to a quantum algorithm with k queries and error $\epsilon' < 1/2$, with ϵ and ϵ' independent of k.

However, there may be a transformation from polynomials of degree $2k$ to quantum algorithms with k queries, with the error $\epsilon' = g(\epsilon, k)$ of the resulting quantum algorithm depending on k but not on function $f(x_1, \ldots, x_n)$ or the number of variables n.

Theorem 15 implies the following limit on such transformations:

Theorem 16. *There is a sequence of Boolean functions $f^{(1)}, f^{(2)}, \ldots$ such that, for any sequence of quantum algorithms $\mathcal{A}_1, \mathcal{A}_2, \ldots$ computing them with $O(\mathrm{bmdeg}(f_i))$ queries, the probability of correct answer is at most*

$$\frac{1}{2} + O\left(\frac{1}{\mathrm{bmdeg}(f^{(i)})}\right).$$

Proof. Let f be the function from Theorem 15. Then, we have $\mathrm{bmdeg}(f) = \tilde{O}(\sqrt{n})$.

If we have a quantum algorithm $\mathcal{A}$ that computes a function f with a probability of correct answer at least $\frac{1}{2} + \delta$, we can use amplitude estimation [9] to estimate whether $\mathcal{A}$ produces answer $f = 1$ with probability at least $\frac{1}{2} + \delta$ or with probability at most $\frac{1}{2} - \delta$. The standard analysis of amplitude estimation [9] shows that we can obtain an estimate that is correct with probability at least 2/3, with $O(1/\delta)$ repetitions of $\mathcal{A}$. To avoid a contradiction with $Q_\epsilon(f) = \Omega(n)$, we must have

$$\frac{\sqrt{n}}{\delta} = \Omega(n)$$

which implies $\delta = O(\frac{1}{\sqrt{n}})$. □

A result with a weaker bound on the error is, however, possible. For example, it is possible that $\widetilde{\deg}_{1/2-\delta}(f) = 2k$ or $\widetilde{\mathrm{bmdeg}}_{1/2-\delta}(f) = 2k$ implies a quantum algorithm which makes k queries and has the error probability at most $\frac{1}{2} - \Omega(\frac{\delta}{2^k})$ or at most $\frac{1}{2} - \Omega(\frac{\delta}{k^2})$.

5 Conclusions

We have shown a new equivalence between quantum algorithms and polynomials: the existence of a 1-query quantum algorithm computing a partial Boolean function f is equivalent to the existence of a degree-2 polynomial p that approximates f. Our equivalence theorem can be seen as a counterpart of the equivalence between unbounded-error quantum algorithms and threshold polynomials, proved by Montanaro et al. [19], and the equivalence between nondeterministic quantum algorithms and nondeterministic polynomials, proved by de Wolf [25].

Our equivalence is, however, much more challenging to prove. A transformation from polynomials to unbounded-error or nondeterministic quantum algorithms can incur a very large loss in error probability (for example, it can transform a polynomial p with error $1/3$ to a quantum algorithm $\mathcal{A}$ with the probability of correct answer $\frac{1}{2} + \frac{1}{2^n}$). In contrast, our transformation produces a quantum algorithm whose error probability only depends on the approximation error of the polynomial p and not on the number of variables n. To achieve this, we use a relation between two matrix norms related to Groethendieck's inequality.

Our equivalence holds for two notions of approximability by a polynomial: the standard one [20] which allows arbitrary polynomials of degree 2 and the approximation by block-multilinear polynomials recently introduced by [1]. The first notion of approximability is known not to be equivalent to the existence of a quantum algorithm: there are several constructions of f for which $Q_\epsilon(f)$ is asymptotically larger than $\deg(f)$ [5, 2], with $Q_\epsilon(f) = \tilde{\Omega}(\deg^2(f))$ as the biggest currently known gap [2]. We have shown that a similar gap holds for the second notion of approximability. Thus, neither of the two notions is equivalent to the existence of a quantum algorithm in the general case.

Three open problems are:

1. **Equivalence between quantum algorithms and polynomials for more than 1 query?**

   Is it true that quantum algorithms with 2 queries are equivalent to polynomials of degree 4? It is even possible that quantum algorithms with k queries are equivalent to polynomials of degree $2k$ for any constant k - as long as the relation between the error of quantum algorithm and the error of the polynomial approximation depends on k, as discussed in section 4.2.

2. **From polynomials to quantum algorithms.**

   It would also be interesting to have more results about transforming polynomials into quantum algorithms, even if such results fell short of a full equivalence between the two notions. For example, if it was possible to transform polynomials of degree 3 into 2 query quantum algorithms this would be an interesting result, even though it would be short of being an equivalence (since 2 query quantum algorithms are transformable into polynomials of degree 4 and not 3).

3. **Other notions of approximability by polynomials?**

   Until this work, there was a hope that the block-multilinear polynomial degree $\widetilde{\mathrm{bmdeg}}(f)$ may provide a quite tight characterization of the quantum query complexity $Q(f)$. Now, we know that the gap between $\mathrm{bmdeg}(f)$ and $Q(f)$ can be as large as the best known gap between $\deg(f)$ and $Q(f)$. Can one come up with a different notion of polynomial degree that would be closer to $Q(f)$ than $\deg(f)$ or $\mathrm{bmdeg}(f)$?

References

[1] S. Aaronson, A. Ambainis. Forrelation: A Problem that Optimally Separates Quantum from Classical Computing. *Proceedings of STOC'2015*, pp. 307-316. Also arxiv:1411.5729.

[2] S. Aaronson, S. Ben-David, R. Kothari. Separations in query complexity using cheat sheets. arXiv:1511.01937.

[3] S. Aaronson, Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4): 595-605, 2004.

[4] A. Acin, N Gisin, B Toner. Grothendieck's constant and local models for noisy entangled quantum states *Physical Review A*, 73 (6), 062105, 2006. Also quant-ph/0606138.

[5] A. Ambainis. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2):220-238, 2006. Earlier versions at FOCS'03 and quant-ph/0305028.

[6] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1): 210-239, 2007. Also FOCS'04 and quant-ph/0311001.

[7] A. Ambainis, A. Childs, B. Reichardt, R. Spalek, S. Zhang. Any AND-OR Formula of Size N Can Be Evaluated in Time $N^{1/2+o(1)}$ on a Quantum Computer. *SIAM Journal on Computing*, 39(6): 2513-2530, 2010. Also FOCS'07.

[8] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778-797, 2001. Earlier versions at FOCS'98 and quant-ph/9802049.

[9] G. Brassard, P. Høyer, M. Mosca, A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information Science*, AMS Contemporary Mathematics Series, 305:53-74, 2002. Also quant-ph/0005055.

[10] M. Braverman, K. Makarychev, Y. Makarychev, A. Naor. The Grothendieck Constant is Strictly Smaller than Krivine's Bound. *Proceedings of FOCS'2011*, pp. 453-462.

[11] H. Buhrman, R. Cleve, J. Watrous, R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16): 167902, 2001. Also quant-ph/0102001.

[12] H. Buhrman, R. de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288:21-43, 2002.

[13] J. Clauser, M. Horne, A. Shimony, R. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23(15):880, 1969.

[14] E. Farhi, J. Goldstone, S. Gutmann, A Quantum Algorithm for the Hamiltonian NAND Tree. *Theory of Computing*, 4:169-190, 2008. Also quant-ph/0702144.

[15] R. Graham, D. Knuth, O. Patashnik. *Concrete Mathematics*, 2nd edition. Addison-Wesley, 1994.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. *Proceedings of STOC'96*, pp. 212-219. Also quant-ph/9605043.

[17] V. Kotesovec. Interesting asymptotic formulas for binomial sums. `http://members.chello.cz/kotesovec/math_articles/kotesovec_interesting_asymptotic_formulas.pdf`, June 2013.

[18] N. Linial, A. Shraibman. Lower bounds in communication complexity based on factorization norms. *Random Structures and Algorithms*, 34(3):368–394, 2009.

[19] A. Montanaro, H. Nishimura, R. Raymond. Unbounded error quantum query complexity. *Theoretical Computer Science*, 412(35):4619-4628, 2011. Also arXiv:0712.1446.

[20] N. Nisan, M. Szegedy. On the Degree of Boolean Functions as Real Polynomials. *Computational Complexity*, 4: 301-313, 1994.

[21] G. Pisier. Grothendieck's theorem, past and present. *Bull. Am. Math. Soc., New Ser.*, 49(2):237–323, 2012.

[22] B. Reichardt. Span-program-based quantum algorithm for evaluating unbalanced formulas. *Proceedings of TQC'2011*, pp. 73-103. Also arXiv:0907.1622.

[23] P. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *SIAM Journal on Computing*, 26:1484-1509, 1997. Also FOCS'94 and quant-ph/9508027.

[24] E. Thomas. A polarization identity for multilinear maps. *Indagationes Mathematicae*, 25: 468-474, 2014. Also arXiv:1309.1275.

[25] R. de Wolf. Nondeterministic quantum query and quantum communication complexities. *SIAM Journal on Computing*, 32(3):681-699, 2003. Also arXiv:cs/0001014.

Appendix

A Proof of Lemma 7

A.1 Additional Notation

The variables of the polynomial (1) correspond to rows and columns of the coefficient matrix $A = (a_{ij})$. Hence, we can reword variable-splitting in terms of rows and columns of A, introducing the operations of *row-splitting* and *column-splitting*.

Let $a_{i\cdot}$ stand for the ith row $(a_{i1}, \ \ldots, \ a_{im})$ of A and similarly $a_{\cdot j}$ stand for the jth column of A. Row-splitting (into k rows) takes a row $a_{i\cdot}$ and replaces it with k equal rows $a_{i\cdot}/k = (a_{i1}/k, \ldots, a_{im}/k)$. Similarly, column-splitting takes a column $a_{\cdot j}$ and replaces it with k equal columns $a_{\cdot j}/k$.

We also denote

$$\|A\|_G = \sup_{k\in\mathbb{N}} \sup_{\substack{p_i,q_j\in\mathbb{R}^k \\ \forall i:\|p_i\|=1 \\ \forall j:\|q_j\|=1}} \sum_{i,j} a_{ij} \, \langle p_i, q_j \rangle.$$

Notice that $\|\cdot\|_G$ is a norm (and, in fact, it is the dual norm of the factorization norm γ_2, see, e.g., [18]).

Let $\lambda_{\max}(B)$ denote the maximal eigenvalue of a square matrix B; then

$$\|A\|^2 = \lambda_{\max}\left(A^T A\right) = \lambda_{\max}\left(AA^T\right). \tag{3}$$

Suppose that $A = (a_{ij})$, $i \in [n]$, $j \in [m]$. Denote

$$g(A) = \frac{\|A\| \sqrt{nm}}{\|A\|_{\infty\to 1}}.$$

By $\Gamma(A)$ we denote the numerator $\|A\| \sqrt{nm}$.

We say that a matrix A' of size $n' \times m'$ can be obtained from A if there exists a sequence of row and column splittings that transforms A to the matrix A'; if A' can be obtained from A, we denote it by $A \longrightarrow A'$. Moreover, for simplicity we assume that no row or column is split repeatedly, i.e., if a row $a_{i\cdot}$ is split into k rows $a_{i\cdot}/k$, then none of these obtained rows is split again.

By $G(A)$ we denote the infimum of $g(A')$ over all matrices A' which can be obtained from A:

$$G(A) := \inf_{A':A\longrightarrow A'} g(A').$$

We have $g(A) \geq 1$ for all matrices A. (To see this, we observe that $\frac{\|Ax\|_1}{\|x\|_\infty} \leq \frac{\sqrt{n}\|Ax\|_2}{\|x\|_2/\sqrt{m}} = \sqrt{nm}\frac{\|Ax\|_2}{\|x\|_2}$. Taking maximums over all x on both sides gives $\|A\|_{\infty\to 1} \leq \sqrt{nm}\|A\|$ which is equivalent to $g(A) \geq 1$.) Therefore, we also have $G(A) \geq 1$.

It is possible to show that the assumption that no row or column is split repeatedly does not alter the value of this infimum; more generally, one could consider weighted splitting of rows (or columns), e.g., allowing to replace a row $a_{i\cdot}$ with k rows $w_j a_{i\cdot}$, $j \in [k]$, where w_j are non-negative weights satisfying $w_1 + \ldots + w_k = 1$. Also in this case it is possible to show that the infimum of $g(A')$ over all matrices A', yielded by permitted splittings, has the same value as $G(A)$.

Let $\mathcal{A}$ denote the class of all matrices (with real entries) which do not contain zero rows or columns. Notice that if $A \in \mathcal{A}$ and $A \longrightarrow A'$, then also $A' \in \mathcal{A}$. The class $\mathcal{A}_{n,m}$ contains all matrices in $\mathcal{A}$ of size $n \times m$.

By $\mathbb{R}^n_+$ we denote the set of all vectors $w \in \mathbb{R}^n$ such that $w_i > 0$ for all $i \in [n]$.

Using the introduced notation, we can restate Lemma 7:

Lemma 7'. *For every matrix A we have*

$$G(A) = \frac{\|A\|_G}{\|A\|_{\infty\to 1}} \le K. \tag{4}$$

The inequality here is due to Grothendieck's inequality, see, e.g., Theorem 4 of [18]. The remaining part of this section is devoted to proving the equality in (4).

A.2 Splitting preserves the infinity-to-one norm

Here we show that splitting rows or columns does not change the norms $\|\cdot\|_{\infty\to 1}$ and $\|\cdot\|_G$.

Lemma 17. *For every matrix $A \in \mathcal{A}$ and every A' s.t. $A \longrightarrow A'$ we have*

$$\|A\|_{\infty\to 1} = \left\|A'\right\|_{\infty\to 1} \quad \textit{and} \quad \|A\|_G = \left\|A'\right\|_G .$$

Proof. Let a matrix $A \in \mathcal{A}_{n,m}$ be fixed. It is sufficient to show the statement for matrices A' that can be obtained by splitting a row $a_{i\cdot}$ of A into $l+1$ rows $a_{i\cdot}/(l+1)$ (these rows are indexed by i, $\ldots$, $i+l$ in A').

Then

$$\|A\|_{\infty\to 1} = \max_{x:\|x\|_\infty \le 1} \|Ax\|_1 = \max_{x\in\{-1,1\}^n} \|Ax\|_1 = \max_{\substack{x\in\{-1,1\}^n \\ y\in\{-1,1\}^m}} x^T A y.$$

Suppose that $x \in \{-1,1\}^n, y \in \{-1,1\}^m$ are such that $x^T A y = \|A\|_{\infty\to 1}$. Notice that

$$x^T A y = \sum_{k=1}^n x_k a_{k\cdot} y.$$

Let $x' \in \{-1,1\}^{n+l}$ be obtained from x by replacing x_i with $(x_i, x_i, \ldots, x_i)$ (i.e., the component x_i, corresponding to the split row $a_{i\cdot}$, is replicated $l+1$ times) and these components are indexed with i, $\ldots$, $i+l$ in x'. Then

$$(x')^T A' y = \sum_{k=1}^{n+l} x'_k a'_{k\cdot} y = (l+1)\cdot x_i \frac{a_{i\cdot}}{l+1} y + \sum_{k\ne i} x_k a_{k\cdot} y = \sum_{k=1}^n x_k a_{k\cdot} y = \|A\|_{\infty\to 1} .$$

This shows that

$$\left\|A'\right\|_{\infty\to 1} \ge \|A\|_{\infty\to 1} .$$

Suppose that $x \in \{-1,1\}^{n+l}, y \in \{-1,1\}^m$ are such that

$$x^T A' y = \left\|A'\right\|_{\infty\to 1}$$

and the rows $a'_{i'\cdot}$, $i' \in [i\,..\,i+l]$, are the rows $a_{i\cdot}/(l+1)$, obtained from $a_{i\cdot}$.

Let $\tilde{x} \in \mathbb{R}^n$ be such that

$$\tilde{x}_k = \begin{cases} x_k & k = 1, 2, \ldots, i-1, \\ x_{k+l} & k = i+1, i+2, \ldots, n, \\ \frac{x_i + \ldots + x_{i+l}}{l+1}, & k = i. \end{cases}$$

Notice that

$$|\tilde{x}_i| \leq \frac{1}{l+1} \sum_{k=i}^{i+l} |x_k| = 1.$$

Thus $\|\tilde{x}\|_\infty \leq 1$. On the other hand,

$$\tilde{x}^T A y = \sum_{k=1}^{n} \tilde{x}_k a_{k\cdot} y = \frac{\sum\limits_{k \in [i\,..\,i+l]} x_k}{l+1} a_{i\cdot} y + \sum_{k=1}^{i-1} x_k a_{k\cdot} y + \sum_{k=i+1}^{n} x_{k+l} a_{k\cdot} y = \sum_{k=1}^{n+l} x_k a'_{k\cdot} y = \left\|A'\right\|_{\infty \to 1}.$$

Since

$$\|A\|_{\infty \to 1} = \sup_{\substack{x \in \mathbb{R}^n, y \in x \in \mathbb{R}^m, \\ \|x\|_\infty \leq 1, \\ \|y\|_\infty \leq 1}} x^T A y,$$

this implies that

$$\|A\|_{\infty \to 1} \geq \left\|A'\right\|_{\infty \to 1}.$$

Hence the two norms are equal.

Consider the norm

$$\|A\|_G = \sup_{r \in \mathbb{N}} \sup_{\substack{p_k, q_j \in \mathbb{R}^r \\ \forall k: \|p_k\| = 1 \\ \forall j: \|q_j\| = 1}} \sum_{k,j} a_{kj} \langle p_k, q_j \rangle.$$

Let unit vectors p_k, q_j (in $\mathbb{R}^r$ for some $r \in \mathbb{N}$) be fixed, $k \in [n]$, $j \in [m]$. Choose $n+l$ unit vectors as follows:

$$p'_k = \begin{cases} p_k, & k < i, \\ p_{k-l}, & k = i+l+1, \ldots, n+l, \\ p_i, & k \in [i\,..\,i+l]. \end{cases}$$

Then

$$\left\|A'\right\|_G \geq \sum_{k,j} a'_{kj} \left\langle p'_k, q_j \right\rangle = \sum_{k,j} a_{kj} \langle p_k, q_j \rangle.$$

Taking the supremum over all r and unit vectors p_k, q_j, we obtain

$$\left\|A'\right\|_G \geq \|A\|_G.$$

Let unit vectors p_k, q_j (in $\mathbb{R}^r$ for some $r \in \mathbb{N}$) be fixed, $k \in [n+l]$, $j \in [m]$.

Choose n unit vectors as follows:

$$\tilde{p}_k = \begin{cases} p_k, & k < i, \\ p_{k+l}, & k = i+1, \ldots, n, \\ \frac{p_i + \ldots + p_{i+l}}{l+1}, & k = i. \end{cases}$$

By the triangle inequality

$$\|\tilde{p}_i\| \le \frac{\|p_i\| + \ldots + \|p_{i+l}\|}{l+1} = 1.$$

Since

$$\|A\|_G = \sup_{r\in\mathbb{N}} \sup_{\substack{p_k, q_j \in \mathbb{R}^r \\ \forall k: \|p_k\|=1 \\ \forall j: \|q_j\|=1}} \sum_{k,j} a_{kj} \langle p_k, q_j \rangle = \sup_{r\in\mathbb{N}} \sup_{\substack{p_k, q_j \in \mathbb{R}^r \\ \forall k: \|p_k\|\le 1 \\ \forall j: \|q_j\|\le 1}} \sum_{k,j} a_{kj} \langle p_k, q_j \rangle,$$

we have

$$\sum_k \sum_j a_{kj} \langle \tilde{p}_k, q_j \rangle \le \|A\|_G .$$

It follows that

$$\sum_{k,j} a'_{kj} \langle p_k, q_j \rangle = \sum_{k\notin[i\,..\,i+l]} \sum_j a'_{kj} \langle p_k, q_j \rangle + \frac{1}{l+1} \sum_{k=i}^{i+l} \sum_j a_{ij} \langle p_k, q_j \rangle = \sum_k \sum_j a_{kj} \langle \tilde{p}_k, q_j \rangle \le \|A\|_G .$$

Taking the supremum over all r and p_k, q_j, we obtain

$$\|A\|_G \ge \|A'\|_G .$$

Hence the two norms are equal.

$\square$

A.3 Characterization of row(column)-splitting

Lemma 18. *Suppose that $A \in \mathcal{A}_{n,m}$; for each $i \in [n]$ the row $a_{i\cdot}$ is split into k_i rows and for each $j \in [m]$ the column $a_{\cdot j}$ is split into l_j rows; the resulting matrix is denoted by A'.*

Then $\Gamma(A') = \|\tilde{A}\| \|w\| \|v\|$, where $\tilde{A} = (\tilde{a}_{ij})$,

$$\tilde{a}_{ij} = \frac{a_{ij}}{w_i v_j},\ i \in [n],\ j \in [m],$$
$$w_i = \sqrt{k_i}, \quad v_j = \sqrt{l_j}.$$

Proof. The matrix A' is of size $(k_1 + \ldots + k_n) \times (l_1 + \ldots + l_m) = \|w\|^2 \|v\|^2$. Hence it is sufficient to show that $\|A'\| = \|\tilde{A}\|$.

We begin by showing this statement in case when $l_1 = l_2 = \ldots = l_m = 1$, i.e., only row-splitting takes place.

Denote $M_i = a_{i\cdot}^T a_{i\cdot}$. By (3),

$$\|\tilde{A}\|^2 = \lambda_{\max}(\tilde{A}^T \tilde{A}), \qquad \|A'\|^2 = \lambda_{\max}(A'^T A'),$$

Notice that

$$\tilde{A}^T \tilde{A} = \begin{pmatrix} w_1^{-1} a_{1\cdot}^T & w_2^{-1} a_{2\cdot}^T & \ldots & w_n^{-1} a_{n\cdot}^T \end{pmatrix} \begin{pmatrix} w_1^{-1} a_{1\cdot} \\ w_2^{-1} a_{2\cdot} \\ \ldots \\ w_n^{-1} a_{n\cdot} \end{pmatrix} = \sum_{i=1}^n w_i^{-2} M_i.$$

Similarly it can be obtained that

$$A'^T A' = \sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{1}{k_i^2} M_i.$$

Since

$$\sum_{i=1}^{n} \sum_{j=1}^{k_i} \frac{1}{k_i^2} M_i = \sum_{i=1}^{n} \frac{1}{k_i} M_i = \sum_{i=1}^{n} w_i^{-2} M_i,$$

we conclude that

$$A'^T A' = \tilde{A}^T \tilde{A},$$

which implies $\left\|\tilde{A}\right\| = \|A'\|$.

Now consider the case of arbitrary $l_j \in \mathbb{N}$. Denote by B the $n \times (l_1 + \ldots + l_m)$ matrix, obtained from A by splitting each of its columns $a_{\cdot j}$ into l_j columns. Then $A \longrightarrow B \longrightarrow A'$. By the previous arguments,

$$\|A'\| = \left\|\tilde{B}\right\|,$$

where $\tilde{B}$ is $\tilde{B}$ is $n \times (l_1 + \ldots + l_m)$ matrix with ith row equal to

$$\left(\underbrace{\frac{a_{i1}}{l_1\sqrt{k_i}}}_{\text{repeated } l_1 \text{ times}} \quad \underbrace{\frac{a_{i2}}{l_2\sqrt{k_i}}}_{\text{repeated } l_2 \text{ times}} \quad \cdots \quad \underbrace{\frac{a_{im}}{l_m\sqrt{k_i}}}_{\text{repeated } l_m \text{ times}} \right).$$

Then the transpose of $\tilde{B}$ can be obtained from the $m \times n$ matrix $C = (C_{ji})$,

$$C_{ji} = \frac{a_{ji}}{\sqrt{k_i}}, \quad i \in [n],\ j \in [m],$$

by splitting the jth row of C into l_j rows.

By previous argument,

$$\left\|\tilde{B}^T\right\| = \left\|\tilde{C}\right\|,$$

where $\tilde{C} = \tilde{A}^T$. Thus we conclude

$$\|A'\| = \left\|\tilde{B}\right\| = \left\|\tilde{B}^T\right\| = \left\|\tilde{A}^T\right\| = \left\|\tilde{A}\right\|.$$

□

This shows that $\Gamma(A')$, for every matrix A' which can be obtained from A by splitting rows/columns, can be characterized by vectors w, v (s.t. the squares of components of w, v are rational numbers). The converse is also true:

Lemma 19. *Suppose that $A \in \mathcal{A}_{n,m}$ but vectors $w \in \mathbb{R}_+^n$, $v \in \mathbb{R}_+^m$ are such that $w_i^2 \in \mathbb{Q}$, $v_j^2 \in \mathbb{Q}$ for all i, j. Then there exist numbers $k_i \in \mathbb{N}$ and $l_j \in \mathbb{N}$ such that splitting A's ith row $a_{i\cdot}$ into k_i rows and the jth column $a_{\cdot j}$ into l_j rows yields a matrix A' such that $\Gamma(A') = \left\|\tilde{A}\right\| \|w\| \|v\|$ where $\left\|\tilde{A}\right\| = (\tilde{a}_{ij})$, $\tilde{a}_{ij} := \frac{a_{ij}}{w_i v_j}$.*

Proof. First note that the statement is true if $w_i^2 \in \mathbb{N}$ and $v_j^2 \in \mathbb{N}$ for all i, j, since then one takes $k_i = w_i^2$ and $l_j = v_j^2$.

Since $w_i^2 \in \mathbb{Q}$, $v_j^2 \in \mathbb{Q}$, we have $w_i^2 = \frac{p_i}{p_i'}$ and $v_j^2 = \frac{q_j}{q_j'}$ for some natural numbers p_i, p_i', q_j and q_j'. Denote $P = \prod_i p_i'$ and $Q = \prod_j q_j'$. Let $\hat{w}_i = w_i \sqrt{P}$, $\hat{v}_j = v_j \sqrt{Q}$ and $\hat{A} = (\hat{a}_{ij})$, where

$$\hat{a}_{ij} = \frac{a_{ij}}{\hat{w}_i \hat{v}_j} = \frac{\tilde{a}_{ij}}{\sqrt{PQ}}.$$

Then

$$\left\|\hat{A}\right\| = \frac{1}{\sqrt{PQ}} \left\|\tilde{A}\right\|, \quad \|\hat{w}\| = \sqrt{P}\,\|w\|, \quad \|\hat{v}\| = \sqrt{Q}\,\|v\|.$$

Thus

$$\left\|\tilde{A}\right\| \|w\| \|v\| = \left\|\hat{A}\right\| \|\hat{w}\| \|\hat{v}\|.$$

Moreover, $\hat{w}_i^2 \in \mathbb{N}$, $\hat{v}_j^2 \in \mathbb{N}$, thus one can take $k_i = \hat{w}_i^2$ and $l_j = \hat{v}_j^2$. Now, by performing the corresponding row/column splitting, one obtains a matrix A' satisfying

$$\Gamma(A') = \left\|\hat{A}\right\| \|\hat{w}\| \|\hat{v}\| = \left\|\tilde{A}\right\| \|w\| \|v\|.$$

$\square$

We can consider an even more general situation:

Lemma 20. *Suppose that $A \in \mathcal{A}_{n,m}$ and $w \in \mathbb{R}_+^n$, $v \in \mathbb{R}_+^m$.*

Then there exist sequences $(k_{i,N})_N \subset \mathbb{N}$ and $(l_{j,N})_N \subset \mathbb{N}$ such that

$$\lim_{N\to\infty} \Gamma(A_N') = \left\|\tilde{A}\right\| \|w\| \|v\|.$$

Here by $\tilde{A}$ we denote the matrix with components $\tilde{a}_{ij} = \frac{a_{ij}}{w_i v_j}$, but A_N' stands for the matrix which is obtained from A by splitting its ith row $a_{i\cdot}$ into $k_{i,N}$ rows and the jth column $a_{\cdot j}$ into $l_{j,N}$ rows.

Proof. We choose two sequences of vectors $w^{(1)}, w^{(2)}, \ldots$ and $v^{(1)}, v^{(2)}, \ldots$ so that $w^{(N)} \in Q_+^n$ and $w = \lim_{N\to\infty} w^{(N)}$ and similarly for $v^{(N)}$ and v. Let $\tilde{A}^{(N)}$ be a matrix with entries $\tilde{a}_{ij}^{(N)} = \frac{a_{ij}}{w_i v_j}$.

Then, by Lemma 19, there are matrices A_N' such that $\Gamma(A_N') = \|\tilde{A}^{(N)}\| \|w^{(N)}\| \|v^{(N)}\|$. Let $k_{i,N}$ and $l_{i,N}$ be the values of k_i and l_i in the application of Lemma 19. By continuity, if $N \to \infty$, we have $\|w^{(N)}\| \to \|w\|$, $\|v^{(N)}\| \to \|v\|$, $\|\tilde{A}^{(N)}\| \to \|\tilde{A}\|$.

Hence, $\lim_{N\to\infty} \Gamma(A_N') = \left\|\tilde{A}\right\| \|w\| \|v\|$. $\square$

Suppose that $A \in \mathcal{A}_{n,m}$ and $w \in \mathbb{R}_+^n$, $v \in \mathbb{R}_+^m$ are fixed. Let $\tilde{A}$ be the matrix with components

$$\tilde{a}_{ij} = \frac{a_{ij}}{w_i v_j}.$$

Notice that $\tilde{A} = D_w^{-1} A D_v^{-1}$. Denote

$$F_A(w, v) = \left\|D_w^{-1} A D_v^{-1}\right\| \|w\| \|v\|.$$

Then Claims 18 and 20 together imply that

$$\inf_{A':A\longrightarrow A'} \Gamma(A') = \inf_{\substack{w\in\mathbb{R}^n_+\\ v\in\mathbb{R}^m_+}} F_A(w,v).$$

Denote the latter infimum with F_A^T. In view of Lemma 17 this means that

$$G(A) = \frac{\inf_{A':A\longrightarrow A'} \Gamma(A')}{\|A\|_{\infty\to 1}} = \frac{F_A^T}{\|A\|_{\infty\to 1}}. \tag{5}$$

A.4 Proof of Lemma 7'

We recall the following characterization of matrices with $\|A\|_G \le 1$; for a proof, see [21, p. 239].

Lemma 21. *For every matrix A (of size $n\times n$), the inequality $\|A\|_G \le 1$ holds iff there is a matrix $\tilde{A}$ (of size $n\times n$) and vectors $w, v \in \mathbb{R}^n$ with non-negative components s.t. $\|w\| = \|v\| = 1$, $\left\|\tilde{A}\right\| \le 1$ and for all $i, j \in [n]$*

$$a_{ij} = \tilde{a}_{ij} w_i v_j.$$

From this it is easy to obtain the following:

Lemma 22. *For every matrix $A \in \mathcal{A}_{n,n}$ there exists a matrix $\tilde{A} \in \mathcal{A}_{n,n}$ and vectors $w, v \in \mathbb{R}^n_+$ s.t. $\|w\| = \|v\| = 1$, $\left\|\tilde{A}\right\| = \|A\|_G$ and $\tilde{A} = D_w^{-1} A D_v$. Moreover, w and v minimize the function $F_A(\cdot,\cdot)$, i.e.,*

$$F_A^T = \left\|\tilde{A}\right\| \|w\| \|v\| = \|A\|_G\,.$$

Proof. Suppose that a matrix $A \in \mathcal{A}_{n,n}$ is scaled so that $\|A\|_G = 1$.

From Lemma 21 the existence of $\tilde{A}$ with $\left\|\tilde{A}\right\| \le 1$ and $w, v \in \mathbb{R}^n_+$ with $\|w\| = \|v\| = 1$ follows. Notice that $w_i \ne 0$ and $w'_j \ne 0$ for all i, j, since otherwise $A \notin \mathcal{A}$. Similarly, also $\tilde{A} \in \mathcal{A}_{n,n}$ must hold.

We claim that $\left\|\tilde{A}\right\| = 1$. Assume the contrary, $\left\|\tilde{A}\right\| = c \in (0,1)$.

Let $\tilde{B}$ be a $n \times n$ matrix with

$$\tilde{b}_{ij} = \tilde{a}_{ij}/c,$$

then $\left\|\tilde{B}\right\| = 1$ and by Lemma 21 we have $\|B\|_G \le 1$, where $B = A/c$. But then

$$\|A\|_G \le c < 1,$$

a contradiction. Thus $\left\|\tilde{A}\right\|_G = 1$.

To prove the second part of the statement, suppose that there are unit vectors $\hat{w}, \hat{v} \in \mathbb{R}^n_+$ such that $F_A(\hat{w}, \hat{v}) = s < 1$. Let $\tilde{X} = D_{\hat{w}}^{-1} A D_{\hat{v}}^{-1}/s$, then $\left\|\tilde{X}\right\| = 1$. By Lemma 21 we have $\|X\|_G \le 1$, where $X = A/s$. But then $\|A\|_G \le s < 1$, a contradiction. $\square$

Proof of Lemma 7'. **The case of** $A \in \mathcal{A}$.

Notice that

$$\inf_{A':A \longrightarrow A'} \Gamma(A') = \inf_{A':A'' \longrightarrow A'} \Gamma(A'),$$

where A'' is any matrix s.t. $A \longrightarrow A''$. This means that

$$F_A^T = F_{A'}^T, \quad \text{if } A \longrightarrow A'.$$

To apply Lemma 22, transform A into a square matrix A' by splitting a row or a column. Then

$$F_A^T = F_{A'}^T \overset{\text{Lemma 22}}{=} \left\|A'\right\|_G \overset{\text{Lemma 17}}{=} \|A\|_G$$

and, by (5),

$$G(A) = \frac{\|A\|_G}{\|A\|_{\infty\to 1}},$$

proving (4) for all $A \in \mathcal{A}$.

It remains to show that (4) holds for all matrices A.

The case of $A \notin \mathcal{A}$. Suppose that A is a $n \times m$ matrix and there are k zero rows and l zero columns. W.l.o.g. assume the non-zero rows/columns are the first, then

$$A = \begin{pmatrix} \hat{A} & 0_{n-k,l} \\ 0_{k,m-l} & 0_{k,l} \end{pmatrix},$$

where $\hat{A} \in \mathcal{A}_{n-k,m-l}$ (and $0_{a,b}$ stands for the zero matrix of size $a \times b$). Notice that

$$g(\hat{A}) = \frac{\left\|\hat{A}\right\| \sqrt{(n-k)(m-l)}}{\left\|\hat{A}\right\|_{\infty\to 1}} = \frac{\|A\| \sqrt{(n-k)(m-l)}}{\|A\|_{\infty\to 1}} < \frac{\|A\| \sqrt{nm}}{\|A\|_{\infty\to 1}} = g(A).$$

By the previous case, we have

$$G(\hat{A}) = \frac{\left\|\hat{A}\right\|_G}{\left\|\hat{A}\right\|_{\infty\to 1}} = \frac{\|A\|_G}{\|A\|_{\infty\to 1}}.$$

Clearly, for every A' with $A \longrightarrow A'$ we have $\hat{A}'$ s.t. $\hat{A} \longrightarrow \hat{A}'$ and $g(\hat{A}') \le g(A')$ (take $\hat{A}'$ to be the minor of A', obtained by skipping all zero rows or columns). Then

$$G(\hat{A}) \le g(\hat{A}') < g(A').$$

Taking infimum over all A' s.t. $A \longrightarrow A'$, inequality $G(\hat{A}) \le G(A)$ follows.

On the other hand, for every $\hat{A}'$ s.t. $\hat{A} \longrightarrow \hat{A}'$ we have a sequence $(A_N)_{N\in\mathbb{N}}$ with $A \longrightarrow A_N$ for all N and $\lim_{N\to\infty} g(A_N) = g(\hat{A}')$: take the matrix

$$B = \begin{pmatrix} \hat{A}' & 0_{p,l} \\ 0_{k,q} & 0_{k,l} \end{pmatrix},$$

where $\hat{A}'$ is of size $p \times q$ (i.e., B is the matrix obtained by splitting the non-zero part of A in the same way how we split $\hat{A}$ to obtain $\hat{A}'$). Then the matrix A_N is obtained by splitting each row $b_{i\cdot}$,

$i \in [p]$ of B, and each column $b_{\cdot j}$, $j \in [q]$ of B into N rows/columns. We have $A \longrightarrow B \longrightarrow A_N$ and the resulting matrix A_N is of size $(Np+k) \times (Nq+l)$. We denote the upper $Np \times Nq$ submatrix of A_N by B_N. Then $B_N = \frac{1}{N^2}\hat{A}' \otimes J_{N,N}$, where $J_{N,N}$ is the $N \times N$ all-1 matrix.

We have

$$\begin{aligned}
&\|A_N\| = \|B_N\| = \frac{\left\|\hat{A}'\right\|}{N};\\
&\|A_N\|_{\infty\to 1} = \|B_N\|_{\infty\to 1} = \left\|\hat{A}'\right\|_{\infty\to 1};\\
&g(A_N) = \frac{\|A_N\|\sqrt{(Np+k)\cdot(Nq+l)}}{\|A_N\|_{\infty\to 1}} = \frac{\|B_N\|\sqrt{(Np+k)\cdot(Nq+l)}}{\|B_N\|_{\infty\to 1}}\\
&= \frac{\left\|\hat{A}'\right\|\sqrt{pq}}{\left\|\hat{A}'\right\|_{\infty\to 1}} \cdot \sqrt{\frac{Np+k}{Np}\cdot\frac{Nq+l}{Nq}} = g(\hat{A}')\sqrt{\left(1+\frac{c_1}{N}\right)\left(1+\frac{c_2}{N}\right)},
\end{aligned}$$

where $c_1 = k/p$, $c_2 = l/q$.

We see that

$$G(A) \le \lim_{N\to\infty} g(A_N) = g(\hat{A}').$$

Taking infimum over all $\hat{A}'$ s.t. $\hat{A} \longrightarrow \hat{A}'$, inequality $G(\hat{A}) \ge G(A)$ follows. Hence the two quantities must be equal. $\square$

B Proof of Lemma 11

B.1 Proof overview

Equivalently, we can construct a block-multilinear polynomial $h : \mathbb{R}^{d(n+1)} \to \mathbb{R}$ which satisfies the following equality

$$h(1, x_1, \ldots, x_n, 1, x_1, \ldots, x_n, \ldots, 1, x_1, \ldots, x_n) = g(x_1, \ldots, x_n)$$

for all $x_1, \ldots, x_n \in \mathbb{R}$ and $|h(y)| \le B(d)$ for all $y \in \{-1,1\}^{d(n+1)}$.

We expand the polynomial g in the Fourier basis as

$$g(x) = \sum_{\substack{T\subset[n]:\\ |T|\le d}} \hat{g}_T \chi_T(x),$$

where $\chi_T(x) = \prod_{i\in T} x_i$. For each $\chi_T(x)$, we define a corresponding block multilinear polynomial

$$\chi'_T\left(z^{(1)}, z^{(2)}, \ldots, z^{(d)}\right) = \sum_{\substack{B\subset[d]:\\ |B|=|T|}} \sum_{\substack{b:\\ b:B\to T\\ b-\text{bijection}}} \frac{(d-r)!}{d!} \prod_{j\in B} z^{(j)}_{b(j)} \prod_{k\in[d]\setminus B} z^{(k)}_0.$$

where $r = |T|$. We then take

$$h\left(z^{(1)}, z^{(2)}, \ldots, z^{(d)}\right) = \sum_T \hat{g}_T \chi'_T\left(z^{(1)}, z^{(2)}, \ldots, z^{(d)}\right).$$

If we set

$$\hat{z}_k^{(j)} = \begin{cases} 1, & k = 0; \\ x_k, & k \in [n] \end{cases}$$

for some $x \in \mathbb{R}^n$, we get

$$\chi'_T\left(\hat{z}^{(1)}, \hat{z}^{(2)}, \ldots, \hat{z}^{(d)}\right) = \sum_{\substack{B \subset [d]: \\ |B| = |T|}} \sum_{\substack{b: \\ b: B \to T \\ b - \text{bijection}}} \frac{(d-r)!}{d!} \prod_{j \in B} x_{b(j)} = \binom{d}{r} r! \, \frac{(d-r)!}{d!} \prod_{j \in T} x_s = \chi_T(x)$$

and, therefore, $h(1, x, 1, x, \ldots, 1, x) = g(x)$.

Since h is multilinear, its maximum over $z^{(j)} \in [-1;1]^{n+1}$, $j \in [d]$, coincides with its maximum over $z^{(j)} \in \{-1, 1\}^{n+1}$, $j \in [d]$. Moreover, we can assume that $z_0^{(j)} = 1$ for all $j \in [d]$. (If $z_0^{(j)} = -1$, we multiply all $z_i^{(j)}$ by -1 and $|h\left(z^{(1)}, z^{(2)}, \ldots, z^{(d)}\right)|$ stays unchanged.) Therefore, if we define

$$h'\left(x^{(1)}, x^{(2)}, \ldots, x^{(d)}\right) = h\left(1, x^{(1)}, 1, x^{(2)}, \ldots, 1, x^{(d)}\right),$$

the maximum of $|h\left(z^{(1)}, \ldots, z^{(d)}\right)|$ over $z^{(j)} \in \{-1, 1\}^{n+1}$, $j \in [d]$ is the same as the maximum of $|h'\left(x^{(1)}, \ldots, x^{(d)}\right)|$ over $x^{(j)} \in \{-1, 1\}^n$, $j \in [d]$.

We have $h'\left(x^{(1)}, \ldots, x^{(d)}\right) = \sum_T \hat{g}_T \chi''_T\left(x^{(1)}, \ldots, x^{(d)}\right)$ where

$$\chi''_T\left(x^{(1)}, x^{(2)}, \ldots, x^{(d)}\right) = \sum_{\substack{B \subset [d]: \\ |B| = |T|}} \sum_{\substack{b: \\ b: B \to T \\ b - \text{bijection}}} \frac{(d-r)!}{d!} \prod_{j \in B} x_{b(j)}^{(j)}.$$

In section B.2, we show

Lemma 23. *For all $u^{(1)}$, ..., $u^{(m)} \in \mathbb{R}^n$ and all $T \subseteq [n]$, $|T| \leq m$, we have*

$$\chi''_T\left(u^{(1)}, u^{(2)}, \ldots, u^{(m)}\right) = \frac{1}{m!} \sum_{\substack{S \subset [m]: \\ S \neq \emptyset}} (-1)^{m-|S|} \; |S|^m \; \chi_T\left(\frac{\sum_{j \in S} u^{(j)}}{|S|}\right) \tag{6}$$

By multiplying (6) with $\hat{g}_T$ and summing over all $T : |T| \leq d$, we get

$$h'\left(x^{(1)}, x^{(2)}, \ldots, x^{(d)}\right) = \frac{1}{d!} \sum_{\substack{S \subset [d]: \\ S \neq \emptyset}} (-1)^{d-|S|} \; |S|^d \; g\left(\frac{\sum_{j \in S} x^{(j)}}{|S|}\right).$$

By taking absolute values, we get

$$\left|h'\left(x^{(1)}, x^{(2)}, \ldots, x^{(d)}\right)\right| \leq \frac{1}{d!} \sum_{\substack{S \subset [d]: \\ S \neq \emptyset}} |S|^d \left|g\left(\frac{\sum_{j \in S} x^{(j)}}{|S|}\right)\right|.$$

For all $x^{(1)}, \ldots, x^{(d)} \in \{-1,1\}^n$ and any nonempty $S \subset [d]$, we have $\frac{\sum_{j\in S} x^{(j)}}{|S|} \in [-1;1]^n$. Since g is multilinear and satisfies $|g(x)| \le 1$ for all $x \in \{-1,1\}^n$, then g also satisfies $|g(x)| \le 1$ for all $x \in [-1;1]^n$. We conclude that the maximum of $\left|h\left(x^{(1)}, x^{(2)}, \ldots, x^{(d)}\right)\right|$ is at most

$$\frac{1}{d!} \sum_{\substack{S\subset[d]:\\ S\neq\emptyset}} |S|^d = \frac{1}{d!} \sum_{s=1}^{d} \binom{d}{s} s^d := B(d).$$

It remains to show that $B(d) = \Theta\left(\frac{\alpha^d}{\sqrt{d}}\right)$. Let $\beta = 1/\alpha = W(1/e)$. It is known [17] that

$$\sum_{s=1}^{d} \binom{d}{s} s^d \sim \frac{1}{\sqrt{1+\beta}} \left(\frac{d}{e\beta}\right)^d.$$

By Stirling's formula,

$$d! \sim \sqrt{2\pi d} \left(\frac{d}{e}\right)^d.$$

Thus

$$B(d) \sim \frac{1}{\sqrt{2\pi(1+\beta)d}} \left(\frac{d}{e\beta}\right)^d \left(\frac{d}{e}\right)^{-d} = \frac{\alpha^d}{\sqrt{2\pi(1+\beta)d}} = \Theta\left(\frac{\alpha^d}{\sqrt{d}}\right).$$

B.2 Proof of Lemma 23

We start with proving two auxiliary lemmas.

Lemma 24. *Suppose that $l, m \in \mathbb{N}$, $l \le m$ and $k \in [0\,..\,m-k]$. Then we have*

$$\sum_{s=l}^{m} \frac{(-1)^{m-s}\, s^k}{(m-s)!\,(s-l)!} = \begin{cases} 0, & k < m-l, \\ 1, & k = m-l. \end{cases}$$

Proof. Let Δ be the difference operator: $\Delta f = f(x+1) - f(x)$, where $f : \mathbb{R} \to \mathbb{R}$. We then have ([15], equation (5.40)):

$$\Delta^n f(x) = \sum_{t=0}^{n} \binom{n}{t} (-1)^{n+t} f(n+t),$$

where $n \in \mathbb{N}$. Apply this to $f(x) = x^k$, where $k \in [0\,..\,n]$ and notice that if $k < n$ then $\Delta^n f = 0$ and if $k = n$ then $\Delta^n f = n!$:

$$\sum_{t=0}^{n} \binom{n}{t} (-1)^{n+t} (x+t)^k = \begin{cases} 0, & k < n \\ n!, & k = n, \end{cases} \quad \text{for all } x \in \mathbb{R}.$$

Multiplying this equality with $\frac{(-1)^{n+k}}{n!}$ yields

$$\sum_{t=0}^{n} \frac{(-1)^t (x-t)^k}{(n-t)!\, t!} = \begin{cases} 0, & k < n \\ 1, & k = n, \end{cases} \quad \text{for all } x \in \mathbb{R}.$$

Now let $n = m - l$, $x = m$, $t = m - s$, then $s = m - t \in [l \mathinner{..} m]$ and we obtain the desired equality:

$$\sum_{s=l}^{m} \frac{(-1)^{m-s} s^k}{(m-s)!\,(s-l)!} = \begin{cases} 0, & k < m - l \\ 1, & k = m - l. \end{cases}$$

$\square$

Lemma 25. *Suppose that $d, m, n \in \mathbb{N}$, $d \le m$ and $d \le n$. Let $\chi : \mathbb{R}^n \to \mathbb{R}$ be defined with*

$$\chi(u_1, \ldots, u_n) = u_1 u_2 \ldots u_d.$$

Then the following identity holds for all $u^{(1)}, \ldots, u^{(m)} \in \mathbb{R}^n$:

$$\sum_{\substack{T \subset [m]: \\ T \neq \emptyset \\ |T| \le d}} \frac{(m - |T|)!}{(d - |T|)!} (-1)^{d-|T|} \chi\left(\sum_{j \in T} u^{(j)}\right) = \sum_{\substack{S \subset [m]: \\ S \neq \emptyset}} (-1)^{m-|S|} \, |S|^{m-d} \, \chi\left(\sum_{j \in S} u^{(j)}\right).$$

Proof. Fix arbitrary vectors $u^{(1)}, \ldots, u^{(m)} \in \mathbb{R}^n$. Notice that for all nonempty $S \subset [m]$ we have

$$\chi\left(\sum_{j \in S} u^{(j)}\right) = \sum_{\substack{i: \\ i:[d] \to S}} u_1^{(i(1))} u_2^{(i(2))} \cdots u_d^{(i(d))}.$$

It follows that we have to show

$$\sum_{\substack{T \subset [m]: \\ T \neq \emptyset \\ |T| \le d}} \frac{(m - |T|)!}{(d - |T|)!} (-1)^{d-|T|} \sum_{\substack{i: \\ i:[d] \to T}} u_1^{(i(1))} u_2^{(i(2))} \cdots u_d^{(i(d))}$$
$$= \sum_{\substack{S \subset [m]: \\ S \neq \emptyset}} (-1)^{m-|S|} \, |S|^{m-d} \sum_{\substack{b: \\ b:[d] \to S}} u_1^{(b(1))} u_2^{(b(2))} \cdots u_d^{(b(d))}. \qquad (7)$$

We proceed by showing that coefficients at every monomial in the LHS and RHS of (7) are the same.

Fix an arbitrary monomial $u_1^{(l_1)} \ldots u_d^{(l_d)}$. Let $L = \{l_1, \ldots, l_d\} \subset [n]$ (the size of L, of course, may be less than d). For each $S \supset L$ there is a unique map $b : [d] \to S$ such that $b(i) = l_i$ for all $i \in [d]$. On the other hand, if $L \not\subset S$, there is no such map. Hence the coefficient at $u_1^{(l_1)} \ldots u_d^{(l_d)}$ in the RHS of (7) is equal to

$$\sum_{\substack{S \subset [m]: \\ L \subset S}} (-1)^{m-|S|} \, |S|^{m-d}.$$

By similar arguments, the coefficient at $u_1^{(l_1)} \ldots u_d^{(l_d)}$ in the LHS of (7) is equal to

$$\sum_{\substack{T \subset [m]: \\ |T| \le d \\ L \subset T}} \frac{(m - |T|)!}{(d - |T|)!} (-1)^{d-|T|}.$$

Thus we need to show that

$$\sum_{\substack{T\subset[m]:\\ |T|\le d\\ L\subset T}} \frac{(m-|T|)!}{(d-|T|)!}(-1)^{d-|T|} = \sum_{\substack{S\subset[m]:\\ L\subset S}} (-1)^{m-|S|}\ |S|^{m-d}. \tag{8}$$

Let $l=|L|$. It is easy to see that there are exactly $\binom{m-l}{s-l}$ sets S of size s such that $L\subset S\subset[m]$. Similarly, there are exactly $\binom{m-l}{t-l}$ sets T of size t such that $L\subset T\subset[m]$. It follows that (8) is equivalent to

$$\sum_{t=l}^{d} \frac{(m-t)!}{(d-t)!}(-1)^{d-t}\binom{m-l}{t-l} = \sum_{s=l}^{m}(-1)^{m-s}s^{m-d}\binom{m-l}{t-l}, \tag{9}$$

which can be simplified to

$$\sum_{t=l}^{d} \frac{(-1)^{d-t}}{(d-t)!\,(t-l)!} = \sum_{s=l}^{m}(-1)^{m-s}\frac{s^{m-d}}{(m-s)!\,(s-l)!}. \tag{10}$$

By Lemma 24, the RHS of (10) equals

$$\sum_{s=l}^{m}(-1)^{m-s}\frac{s^{m-d}}{(m-s)!\,(s-l)!} = \begin{cases}0, & l<d\\ 1, & l=d.\end{cases}$$

On the other hand, again by Lemma 24,

$$\sum_{t=l}^{d} \frac{(-1)^{d-t}}{(d-t)!\,(t-l)!} = \begin{cases}0, & 0<d-l\\ 1, & 0=d-l.\end{cases}$$

Thus (10) holds, which shows the statement.

□

We now prove lemma 23.

If $T=\emptyset$, the desired equality becomes

$$1 = \frac{1}{m!}\sum_{s=1}^{m}(-1)^{m-s}\binom{m}{s}s^{m}. \tag{11}$$

Since the RHS of (11) is

$$\sum_{s=1}^{m}(-1)^{m-s}\frac{s^{m-1}}{(m-s)!(s-1)!},$$

we see that (11) follows from Lemma 24.

Hence, we can assume that T is non-empty. We now denote $d=|T|$. W.l.o.g. we can suppose that $T=[d]$. For brevity, we denote χ_T as simply χ.

We obviously have that $\chi(cx)=c^d\chi(x)$ for all scalars c. Thus, we have to show the equality

$$\sum_{\substack{J\subset[m]:\\ |J|=d}} \sum_{\substack{b:\\ b:J\to[d]\\ b\text{ – bijection}}} \prod_{j\in J} u^{(j)}_{b(j)} = \frac{1}{(m-d)!}\sum_{\substack{S\subset[m]:\\ S\neq\emptyset}} (-1)^{m-|S|}\ |S|^{m-d}\ \chi\left(\sum_{j\in S}u^{(j)}\right). \tag{12}$$

Fix any $J \subset [m]$ of size d. To simplify the notation we assume that $J = [d]$. We consider $G : \mathbb{R}^{nd} \to \mathbb{R}$ defined by

$$G\left(u^{(1)}, u^{(2)}, \ldots, u^{(d)}\right) = \sum_{\substack{i: \\ i\,:\,[d]\to[d] \\ i\text{ - bijection}}} u_1^{(i(1)))} u_2^{(i(2))} \ldots u_d^{(i(d))}. \tag{13}$$

We can view G as a map $G : E^d \to \mathbb{R}$, where $E = \mathbb{R}^n$, which is

- d-linear: for all $u^{(1)}, \ldots, u^{(d)}, v \in E$, $\alpha, \beta \in \mathbb{R}$ and all $i \in [d]$ we have

$$\begin{aligned} &G\left(u^{(1)}, u^{(2)}, \ldots, u^{(i-1)}, \alpha u^{(i)} + \beta v, u^{(i+1)}, \ldots, u^{(d)}\right) = \\ &\quad = \alpha G\left(u^{(1)}, \ldots, u^{(i-1)}, u^{(i)}, u^{(i+1)}, \ldots, u^{(d)}\right) + \beta G\left(u^{(1)}, \ldots, u^{(i-1)}, v, u^{(i+1)}, \ldots, u^{(d)}\right), \end{aligned}$$

- symmetric: for all bijections $\sigma : [d] \to [d]$ and all $u^{(1)}, \ldots, u^{(d)} \in E$ we have

$$G\left(u^{(1)}, u^{(2)}, \ldots, u^{(d)}\right) = G\left(u^{(\sigma(1))}, u^{(\sigma(2))}, \ldots, u^{(\sigma(d))}\right).$$

As shown by Thomas ([24], Eq. (7) and (3)), we have

Lemma 26. *[24] If G is d-linear and symmetric, we have*

$$d!G\left(u^{(1)}, u^{(2)}, \ldots, u^{(d)}\right) = \sum_{T\subset[d],\ T\neq\emptyset} (-1)^{d-|T|} g\left(\sum_{j\in T} u^{(j)}\right),$$

where $g(x) := G(x, x, \ldots, x)$.

For our choice of G (equation (13)), we have $g(x) = d!\chi(x)$. Therefore,

$$\sum_{\substack{i: \\ i\,:\,[d]\to[d] \\ i\text{ - bijection}}} u_1^{(i(1)))} u_2^{(i(2))} \ldots u_d^{(i(d))} = \sum_{\substack{T\subset[d] \\ T\neq\emptyset}} (-1)^{d-|T|} \chi\left(\sum_{j\in T} u^{(j)}\right).$$

If we perform the same argument with an arbitrary $J : |J| = d$ instead of $J = [d]$, we get

$$\sum_{\substack{i: \\ i\,:\,J\to[d] \\ i\text{ - bijection}}} \prod_{j\in J} u_j^{(i(j)))} = \sum_{\substack{T\subset J \\ T\neq\emptyset}} (-1)^{d-|T|} \chi\left(\sum_{j\in T} u^{(j)}\right).$$

By summing over all J and combining the result with (12), we get that

$$\sum_{\substack{J\subset[m]: \\ |J|=d}} \sum_{\substack{T\subset J \\ T\neq\emptyset}} (-1)^{d-|T|} \chi\left(\sum_{j\in T} u^{(j)}\right) = \frac{1}{(m-d)!} \sum_{\substack{S\subset[m]: \\ S\neq\emptyset}} (-1)^{m-|S|}\ |S|^{m-d}\ \chi\left(\sum_{j\in S} u^{(j)}\right). \tag{14}$$

Let a set $T \subset [m]$ of size $|T| \le d$ be fixed. Then there are $\binom{m-|T|}{d-|T|}$ sets $J \subset [m]$ of size $|J| = d$ satisfying $T \subset J$. Hence (14) can be rewritten as

$$\sum_{\substack{T \subset [m]: \\ T \neq \emptyset \\ |T| \le d}} \binom{m-|T|}{d-|T|} (-1)^{d-|T|} \chi\left(\sum_{j \in T} u^{(j)}\right) = \frac{1}{(m-d)!} \sum_{\substack{S \subset [m]: \\ S \neq \emptyset}} (-1)^{m-|S|}\ |S|^{m-d}\ \chi\left(\sum_{j \in S} u^{(j)}\right). \quad (15)$$

But this equality (after multiplying it by $(m-d)!$) follows from Lemma 25, concluding the proof.

C Proof of Theorem 15

We use the notion of *certificate complexity.* Let C be an assignment of values $C : S \to \{0,1\}$ for some $S \subseteq [n]$. We say that $x = (x_1, \ldots, x_n)$ is *consistent* with C if it satisfies $x_i = C(i)$ for all $i \in S$. We say that C is a *certificate* for f on an input x if x is consistent with C and, for any $y \in \{0,1\}^n$ that is consistent with C, we have $f(y) = f(x)$.

The certificate complexity of f on an input x (denoted by $C(f,x)$) is the smallest $|S|$ in a certificate C for f on the input x. The certificate complexity of f (denoted $C(x)$) is the maximum of $C(f,x)$ over all $x \in \{0,1\}^n$. (For more information on the certificate complexity and its connections to other complexity measures, we refer the reader to the survey by Buhrman and de Wolf [12].)

We use the same function as in the $Q(f) = \tilde{\Omega}(\deg^2(f))$ result of Aaronson et al. [2]. The construction of this function [2] starts by designing a function $g : \{-1,1\}^n \to \{0,1\}$ with $Q(g) = \tilde{\Omega}(n)$ and $C(g) = \tilde{O}(\sqrt{n})$. (We omit the definition of g because $Q(g) = \tilde{\Omega}(n)$ and $C(g) = \tilde{O}(\sqrt{n})$ are the only properties of g that we use.)

Then, they define f as follows:

1. The first $c = 10n \log n$ input variables of f are interpreted as c inputs $x^{(1)} \in \{0,1\}^n, \ldots, x^{(c)} \in \{0,1\}^n$ to the function g.
2. These input variables are followed by 2^c groups of variables $y^{(m)}$, $m \in \{0,1\}^c$, with each group containing $cC(g) \log n$ variables. The content of each $y^{(m)}$ is interpreted as descriptions for c sets $S_1, \ldots, S_c \subseteq [n]$ with $|S_j| = C(g)$. A set S_j is interpreted as a sequence of indices for $C(g)$ variables for the function $g(x^{(j)})$.
3. $f = 1$ if and only if, for some $m \in \{0,1\}^c$, the group $y^{(m)}$ contains descriptions for sets S_i such that, for each $i \in [c]$, the variables $x^{(i)}_j$, $j \in S_i$ form an m_i-certificate.

As shown in [2], f satisfies $Q(f) = \tilde{\Omega}(n)$ and $\deg(f) = \tilde{O}(\sqrt{n})$. A polynomial p of degree $\tilde{O}(\sqrt{n})$ that represents f can be constructed as follows:

1. $p = \sum_{m \in \{0,1\}^c} p_m$;
2. $p_m = \sum_{S_1,\ldots,S_c} p_{m,S_1,\ldots,S_c}$, with the summation over all tuples $(S_1, \ldots, S_c)$ such that, for all $i \in [c]$, S_i is a possible certificate for $g(x) = m_i$;
3. $p_{m,S_1,\ldots,S_c} = q_{m,S_1,\ldots,S_c} \prod_{i=1}^{c} r_{i,m_i,S_i}$;
4. $q_{m,S_1,\ldots,S_c} = 1$ if the contents of $y^{(m)}$ describe sets $S_1, \ldots, S_c$ and $q_{m,S_1,\ldots,S_c} = 0$ otherwise;

5. $r_{i,m_i,S_i} = 1$ if the values of variables $x_j^{(i)}, j \in S_i$ certify that $g(x^{(i)}) = m_i$ and $r_{i,m_i,S_i} = 0$ otherwise.

In the non-block-multilinear case, $q_{m,S_1,\ldots,S_c}$ is the product of $\frac{1+y_i^{(m)}}{2}$'s (for i's where we need $y_i^{(m)} = 1$) and $\frac{1-y_i^{(m)}}{2}$'s (for i's where we need $y_i^{(m)} = -1$). r_{i,m_i,S_i} is constructed similarly, by taking a product of $\frac{1+x_j^{(i)}}{2}$'s and $\frac{1-x_j^{(i)}}{2}$'s for $j \in S_i$, to obtain the condition that x_j^i take the values that are necessary so that x_j^i, $j \in S_i$ certify $g(x^{(i)}) = m_i$.

We now modify this construction to obtain $\mathrm{bmdeg}(f) = \tilde{O}(\sqrt{n})$. Our polynomial has blocks of variables $z^{(i)}$, for $i \in [cC(g)(\log n + 1)]$, with each $z^{(i)}$ consisting of a variable $z_0^{(i)}$, c subblocks $x^{(i,1)}, \ldots, x^{(i,c)}$ and 2^c subblocks $y^{(i,m)}$ for $m \in \{0,1\}^c$.

The structure of the polynomial p stays the same and we only modify the constructions of $q_{m,S_1,\ldots,S_c}$ and r_{i,m_i,S_i}. To construct $q_{m,S_1,\ldots,S_c}$, we use the first $cC(g)\log n$ blocks $z^{(i)}$, taking the value of $y_i^{(m)}$ from the i^{th} block and using $z_0^{(i)}$ instead of 1 in the terms $\frac{1 \pm y_i^{(m)}}{2}$.

To construct r_{i,m_i,S_i}, we use $z^{(k)}$ for $k \in \{(c\log n + (i-1))C(g) + 1, \ldots, (c\log n + i)C(g)\}$ and take r_{i,m_i,S_i} to be the average of the desired product of $\frac{z_0^{(k)} + x_j^{(k,i)}}{2}$'s and $\frac{z_0^{(k)} - x_j^{(k,i)}}{2}$'s over all the ways how one could use one term per block $z^{(k)}$.

It is easy to see that, if all blocks $z^{(i)}$ contain the same assignment z, then $p(z, \ldots, z)$ is the same polynomial as in the non-block-multilinear case and is equal to $f(z)$. We now show that $|p| \leq 1$ for any choice of $z^{(1)}, z^{(2)}, \ldots$ in which all the variables are in $\{-1, 1\}$.

For each m, all polynomials $q_{m,S_1,\ldots,S_c}$ use the same variables $z_0^{(i)}$ and $y_i^{(i,m)}$ and are defined so that, for any choice of values for $z_0^{(i)}$'s and $y_i^{(i,m)}$'s, at most one of $q_{m,S_1,\ldots,S_c}$ is ± 1 and the rest are 0. Let $S_{m,1}, \ldots, S_{m,c}$ be the sets for which $q_{m,S_{m,1},\ldots,S_{m,c}} = \pm 1$ (if such sets exist). Then, $p(z^{(1)}, \ldots, z^{(cC(g)(\log n+1))})$ is equal to the sum

$$\sum_{m \in \{0,1\}^c} a_m \prod_{i=1}^{c} r_{i,m_i,S_{m,i}} \tag{16}$$

for some choice of signs $a_m \in \{-1, 1\}$. We show

Lemma 27. *Let $S_{m,i}$, $m \in \{0,1\}^c$, $i \in [c]$ be such that $S_{m,i}$ is an m_i-certificate for the function g. Then,*

$$\left| \sum_{m \in \{0,1\}^c} a_m \prod_{i=1}^{c} r_{i,m_i,S_{m,i}} \right| \leq 1$$

for any choice of signs $a_m \in \{-1, 1\}$.

Proof. By induction on c. For $c = 1$, this simplifies to

$$-1 \leq a_0 r_{1,0,S_{0,1}} + a_1 r_{1,1,S_{1,1}} \leq 1 \tag{17}$$

when $S_{0,1}$ is a set of variables for a 0-certificate and $S_{1,1}$ is a set of variables for a 1-certificate. Since a 0-certificate and a 1-certificate cannot be true at the same time, there must be $j \in S_{0,1} \cap S_{1,1}$ with x_j taking one value in the 0-certificate and another value in the 1-certificate.

Let p_0 be the probability that, when we choose a block $z^{(i)}$ randomly among the blocks that are used to define r_{1,m_1,S_1}'s, we get the value of $x_j^{(i,1)}$ which matches the 0-certificate. Then, the probability of getting the value that matches the 1-certificate is $1 - p_0$ and we get that $r_{1,0,S_{0,1}} \leq p_0$ and $r_{1,1,S_{1,1}} \leq 1 - p_0$. This implies (17) for any choice of signs $a_0, a_1 \in \{-1, 1\}$.

For $c > 1$, we can use the same argument to show that, for any $m \in \{0,1\}^{c-1}$, we have $r_{c,0,S_{m0}} \leq p_m$ and $r_{c,1,S_{m1}} \leq 1 - p_m$ for some p_m that depends on m. Therefore, the sum of Lemma 27 is upper bounded by

$$\sum_{m \in \{0,1\}^{c-1}} \left(p_m a_{m0} \prod_{i=1}^{c-1} r_{i,m_i,S_{m0,i}} + (1 - p_m) a_{m1} \prod_{i=1}^{c-1} r_{i,m_i,S_{m1,i}} \right).$$

We can express this sum as a probabilistic combination of sums

$$\sum_{m \in \{0,1\}^{c-1}} a_m \prod_{i=1}^{c-1} r_{i,m_i,S_{m,i}} \tag{18}$$

where each $S_{m,i}$ is either $S_{m0,i}$ or $S_{m1,i}$ and each a_m is either a_{m0} or a_{m1}. Each of sums (18) is at most 1 in absolute value by the inductive assumption.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de