Electronic Colloquium on Computational Complexity, Report No. 127 (2026)

Approximating Polynomials for De Morgan Formulas with Optimal Coefficient ℓ_1-Norm Bounds

Yichuan Wang*

July 24, 2026

Abstract

We prove that every De Morgan formula[1] with n leaves has a pointwise 1/3-approximating real polynomial of degree $O(\sqrt{n})$ and coefficient ℓ_1-norm $2^{O(\sqrt{n})}$. The standard approximate-degree theorem for formulas gives the same degree bound, but only yields the weaker coefficient estimate $2^{O(\sqrt{n}\log n)}$.

Our proof constructs, for every formula, a span program[2] with witness size $O(\sqrt{n})$ and, additionally, $O(1)$ entrywise-absolute operator norm for the available-vector matrix. A standard span-program-to-polynomial argument then gives the desired approximating polynomial while preserving coefficient weight. We also show that the coefficient bound is tight up to constants in the exponent: the De Morgan formula for inner product modulo 2 already gives a matching $2^{\Omega(\sqrt{n})}$ lower bound for formulas with at most n leaves, even regardless of the degree of the approximating polynomial.

*University of California at Berkeley. Email: `yichuan-21@berkeley.edu`.

[1] i.e., a formula over AND, OR, and NOT gates.

[2] A span program is a linear-algebraic certificate system specified by a target vector and input-labeled vectors; on an input, it accepts exactly when the target vector lies in the span of the vectors made available by that input.

ISSN 1433-8092

Contents

1 Introduction

1.1 De Morgan Formulas and Formula Lower Bounds

De Morgan formulas are one of the most basic non-uniform models of computation. Throughout the paper, a De Morgan formula means a fan-in-two formula over AND, OR, and NOT gates; XOR gates are not part of the basis. Equivalently, after pushing negations to the leaves, the internal gates are AND and OR, and the leaves are literals. The size of a formula is the number of leaf occurrences. Polynomial-size De Morgan formulas capture non-uniform NC^1, and proving strong explicit lower bounds for them remains a central problem in circuit complexity.

The best-known explicit lower bounds for De Morgan formulas remain near the cubic barrier, with the current explicit constructions coming from shrinkage, average-case, and related formula-lower-bound methods [Hås98, KR13, Tal14, KRT17, Bog18, FMT23]. The classical Khrapchenko method gives a quadratic lower bound for parity [Khr71]. The shrinkage theorem in [Hås98] shows that De Morgan formulas have shrinkage exponent 2, yielding an $\Omega(n^{3-o(1)})$ formula-size lower bound for an explicit function in P. Tal [Tal14] gives a spectral proof of tight quadratic shrinkage, and Filmus, Meir, and Tal [FMT23] extend shrinkage to random projections and obtain $\widetilde{\Omega}(n^3)$ lower bounds for an explicit function in AC^0. Average-case lower bounds with an essentially cubic worst-case scale were developed in [KR13, KRT17], and Bogdanov [Bog18] proves formula lower bounds for small-bias functions.

There is also a depth-lower-bound viewpoint. A fan-in-two formula of depth d has size at most 2^d, so a formula-size lower bound of $n^{3-o(1)}$ implies a depth lower bound of $(3-o(1))\log_2 n$. Conversely, formulas of size s can be balanced to depth $O(\log s)$, with only a polynomial increase in size. The Karchmer–Wigderson connection [KW88] relates formula depth to communication complexity, and the Karchmer–Raz–Wigderson program [KRW95] seeks stronger depth lower bounds through direct-sum phenomena. Subsequent works developed information-complexity, XOR-KRW, strong-composition, and related communication-complexity approaches toward this goal [GMWW14, DM16, MS21, Mei23]. The algorithmic framework of [BW24] gives another route: sufficiently strong algorithms for #SAT of subcubic-size De Morgan formulas would yield faster $\mathsf{Gap\text{-}SAT}$ algorithms for larger formulas and, through the algorithmic method, depth lower bounds beyond the current factor-3 barrier.

1.2 Polynomial Approximations of Formulas

One way to study formulas is through real polynomial approximation. We say that a real polynomial P pointwise approximates a Boolean function F if $|P(x) - F(x)| \le 1/3$ for every Boolean input x. The least possible degree of such a polynomial is the approximate degree of F, a notion originating in the study of Boolean functions as real polynomials [NS94]; see also the survey [BT20].

For De Morgan formulas of size n, the worst-case approximate degree has order $\sqrt{n}$. On the upper-bound side, quantum formula-evaluation algorithms and span programs give degree-$O(\sqrt{n})$ approximating polynomials. In particular, Ambainis et al. [ACR+10] give an $n^{1/2+o(1)}$-query quantum algorithm for size-n AND-OR formulas, and the span-program framework of [Rei09, Rei11, Rei12] gives a clean modern formulation of formula evaluation and the adversary method. The polynomial method [BBC+01], adversary lower bounds [LLS06], and dual-polynomial techniques [She08, BKT21, BDBGK18] form the surrounding toolkit. These results control degree, but they do not by themselves give the coefficient-ℓ_1 guarantee proved here.

1.3 Why Coefficient ℓ_1-Norm Matters

The coefficient ℓ_1-norm in this paper is measured in the standard $\{0,1\}$-monomial basis.[3] Concretely, if $P(x) = \sum_T a_T \prod_{i \in T} x_i$, then $\|P\|_1 := \sum_T |a_T|$. A monomial in this basis is an AND of variables. If the variables being tested are allowed to be literals, or outputs of subformulas and their negations, the same object is an indicator of a partial assignment; we will call such tests partial ANDs.

The basic averaging step is simple and is the reason coefficient weight matters. In this motivational paragraph we use the sign convention for Boolean functions, obtained from the 0/1 convention by mapping 0 to 1 and 1 to -1. Let f be a $\{\pm 1\}$-valued target function, let μ be an input distribution, and suppose that a polynomial, written as a signed combination of partial ANDs,

$$P = \sum_T a_T A_T,$$

satisfies $\mathbf{E}_\mu[fP] \geq 1/3$, where each A_T is a partial AND. Let $L := \sum_T |a_T|$. Then,

$$\frac{1}{3} \leq \sum_T |a_T| \cdot |\mathbf{E}_\mu[f A_T]|.$$

Thus some partial AND satisfies $|\mathbf{E}_\mu[f A_T]| \geq 1/(3L)$. A partial AND on d literals is computed by a size-$O(d)$ De Morgan formula. More generally, if its inputs are outputs, or negated outputs, of subformulas $G_1, \dots, G_d$, then it is computed by a De Morgan formula of size

$$\sum_{j=1}^{d} |G_j|,$$

where negating a subformula does not change its leaf size. Consequently, a low-ℓ_1 polynomial approximation can be converted into a smaller formula-like test with advantage inversely proportional to the coefficient weight, provided the input subformulas are themselves small.

This is the coefficient-weight perspective behind several approximation-to-lower-bound arguments. Tal's work [Tal17] uses approximation to reduce lower bounds for large formulas to strong average-case lower bounds against smaller formulas. At a high level, after decomposing a large formula into a top formula whose inputs are smaller subformulas, one approximates the top computation by a polynomial and expands it into partial ANDs over the top inputs. Each such partial AND becomes an AND of smaller subformulas, hence a smaller De Morgan formula. The averaging step above loses a factor equal to the total coefficient weight. Thus improving $\|P\|_1$ directly improves the advantage of the smaller formula extracted from the polynomial.

Related approximation mechanisms appear elsewhere as well. Kabanets et al. [KKL+21] use approximating-polynomial technology for De Morgan formulas with low-communication leaf gates, where the expansion of the approximating polynomial affects the cost of the simulation. Hatami et al. [HHTT21] explicitly track coefficient ℓ_1-norms when robustifying low-error approximating polynomials for De Morgan formulas in pseudorandomness and average-case lower-bound applications. The algorithmic reduction of [BW24] also uses approximating polynomials for formulas as part of a reduction from Gap-SAT on large formulas to #SAT on smaller formulas. More broadly, approximate coefficient weight in the De Morgan basis is a natural representation measure; see, for example, [CDL26] for exact-versus-approximate questions for De Morgan-basis representations.

[3] The Fourier basis is equivalent for our purposes at degree d, up to a $2^{O(d)}$ factor in coefficient ℓ_1-norm. Indeed, a $\{0,1\}$-monomial has Fourier ℓ_1-norm 1, while a Fourier character $\chi_S(x) = (-1)^{\sum_{i \in S} x_i} = \prod_{i \in S}(1 - 2x_i)$ of degree $|S| \leq d$ has $\{0,1\}$-monomial coefficient norm $3^{|S|} \leq 2^{O(d)}$. Thus a $2^{O(\sqrt{n})}$ bound in one of these bases is a $2^{O(\sqrt{n})}$ bound in the other for degree $O(\sqrt{n})$ polynomials.

1.4 Previously Available Coefficient Bounds

The known upper bound on degree gives a straightforward but weaker coefficient estimate. Let P be a degree-$O(\sqrt{n})$ polynomial that pointwise approximates a Boolean function. Since P is bounded by a constant on the Boolean cube, Parseval gives $\sum_S \widehat{P}(S)^2 = O(1)$ in the Fourier basis. The number of Fourier characters of degree at most $O(\sqrt{n})$ is

$$\sum_{j \le O(\sqrt{n})} \binom{n}{j} = 2^{O(\sqrt{n}\log n)}.$$

By Cauchy–Schwarz, the Fourier ℓ_1-norm is at most $2^{O(\sqrt{n}\log n)}$. Converting a degree-$O(\sqrt{n})$ Fourier character to the $\{0,1\}$-monomial basis costs at most another $2^{O(\sqrt{n})}$ factor, which is absorbed in the same bound. Thus the immediate consequence of the approximate-degree theorem is only

$$\|P\|_1 \le 2^{O(\sqrt{n}\log n)}.$$

There is an unpublished recursive improvement based on input-robust polynomials, communicated to us by Lijie Chen [Che18]. An input-robust approximating polynomial continues to approximate the Boolean function when each input bit is replaced by a real number close to either 0 or 1, for instance, in $[0,1/3]\cup[2/3,1]$. Robust polynomial techniques were developed in [BNRdW07, She13], and coefficient ℓ_1-norms under robustification are tracked explicitly in [HHTT21]. Robustness is useful because robust polynomials compose: if a top formula is robustly approximated and each of its input subformulas is approximated, then substituting the bottom approximants into the top robust polynomial still approximates the whole formula.

A standard decomposition lemma, going back through the formula-decomposition and shrinkage/PRG literature and used in forms such as [IMZ12, Tal14, Tal17, BW24], writes a size-n formula as a top formula of size $O(n/k)$ whose inputs are subformulas of size $O(k)$. Starting from the input-robust coefficient bound $2^{O(\sqrt{s}\log s)}$ for size-s formulas, one approximates the top formula and all bottom subformulas by robust polynomials and composes them. The top polynomial has degree $O(\sqrt{n/k})$. Therefore the coefficient weight after one composition is bounded by

$$2^{O(\sqrt{n/k}\log(n/k))} \cdot \left(2^{O(\sqrt{k}\log k)}\right)^{O(\sqrt{n/k})}.$$

The degree remains $O(\sqrt{n/k}) \cdot O(\sqrt{k}) = O(\sqrt{n})$. Choosing $k = (\log n)^2$ gives coefficient ℓ_1-norm $2^{O(\sqrt{n}\log\log n)}$. Repeating the same decomposition yields bounds involving any fixed number of iterated logarithms. Iterating the process for $\log^* n$ rounds yields a $\log^* n$-type loss: degree $\sqrt{n}\,2^{O(\log^* n)}$ and coefficient norm $2^{O(\sqrt{n}\log^* n)}$ [Che18].

The present paper removes all these logarithmic losses: we prove the optimal coefficient bound $2^{O(\sqrt{n})}$ while keeping degree $O(\sqrt{n})$.

1.5 Main Result and Tightness

Our main theorem is the following. It is restated and proved as Theorem 5.1.

Theorem 1.1 (Main theorem; see also Theorem 5.1)**.** *Let $F : \{0,1\}^m \to \{0,1\}$ be computed by a De Morgan formula with n leaf occurrences. Then there is a real multilinear polynomial P in the m input variables such that, for every $x \in \{0,1\}^m$,*

$$|P(x) - F(x)| \le \frac{1}{3},$$

and

$$\deg P = O(\sqrt{n}), \qquad \|P\|_1 \le 2^{O(\sqrt{n})}.$$

The coefficient bound is optimal, up to the constant in the exponent, for formulas of this size. The obstruction is inner product modulo 2 on $t = \Theta(\sqrt{n})$ pairs of variables. This function has a De Morgan formula with $O(t^2) \le n$ leaves, and its sign version has all Fourier coefficients of magnitude 2^{-t}. Hence every pointwise approximant has Fourier coefficient ℓ_1-norm $2^{\Omega(t)}$. Since each $\{0,1\}$-monomial is an AND and has Fourier ℓ_1-norm exactly 1, the $\{0,1\}$-monomial coefficient norm is at least the Fourier coefficient norm. Thus any pointwise approximant, with no degree restriction, has coefficient ℓ_1-norm $2^{\Omega(\sqrt{n})}$. Theorem 6.1 gives the short argument.

Application – Slightly Improving Formula Lower Bounds Plugging the present coefficient bound into Tal's size-amplification argument [Tal17] improves the quantitative averaging step. Let $L(f)$ denote the minimum size of an exact De Morgan formula for f, and suppose that every formula of size at most s has agreement at most $1/2 + 2^{-\Omega(r)}$ with f under the relevant input distribution. Tal's decomposition, with parameter k, produces a formula of size $O(L(f)/k)$ whose advantage is the reciprocal of the coefficient loss for a top formula with $O(k^2)$ leaves. With the previous coefficient estimate, this loss is $2^{O(k \log k)}$. To contradict the assumed average-case hardness, one needs

$$\frac{L(f)}{k} \lesssim s, \qquad k \lesssim s, \qquad k \log k \lesssim r.$$

Optimizing k gives

$$L(f) = \Omega\left(\min\left\{s^2, \frac{sr}{\log r}\right\}\right),$$

up to absolute constants. Our theorem changes the coefficient loss to $2^{O(k)}$, so the last condition becomes $k \lesssim r$, and the same optimization gives

$$L(f) = \Omega\big(\min\{s^2, sr\}\big).$$

In the parameter setting of Section 4.1 of [Tal17], one has

$$sr = \Theta\left(\frac{n^3}{\log n \, \log\log n}\right), \qquad \log r = \Theta(\log\log n),$$

and the s^2 term is not the bottleneck. Hence the resulting explicit lower bound improves from

$$\Omega\left(\frac{n^3}{\log n \, (\log\log n)^2}\right) \quad \text{to} \quad \Omega\left(\frac{n^3}{\log n \, \log\log n}\right).$$

2 Proof Overview

This section summarizes the proof of Theorem 5.1. The proof has two parts. First, we convert a span program with suitable quantitative properties into an approximating polynomial. Second, we construct such a span program for every De Morgan formula. The second part is the main contribution: the span program must have small witness size, and the full matrix of its listed vectors must also have small entrywise-absolute operator norm.

Throughout the overview, it is helpful to begin with a read-once monotone formula in formal variables associated with the leaf occurrences. Repeated variables and negated literals are handled at the end by replacing each formal leaf variable with the literal x_i or $1 - x_i$ that labels that occurrence. Identifying repeated positive occurrences and multilinearizing do not increase the degree or the coefficient ℓ_1-norm, because these operations only merge monomials. Substituting a negated literal also preserves the degree and increases the coefficient ℓ_1-norm by at most a factor of 2 for each variable in a monomial. Therefore, for a degree-d polynomial, the total loss is at most 2^d, which is absorbed into the final $2^{O(\sqrt{n})}$ bound.

A vector-form span program has a unit target vector and a list of labeled unit vectors, each labeled one of *always available*, x_i, or $\neg x_i$. On an input x, availability is determined by the corresponding coordinate values: an always-available vector is available on every input, an x_i-labeled vector is available when $x_i = 1$, and a $\neg x_i$-labeled vector is available when $x_i = 0$. The span program accepts an input when the target vector lies in the span of the vectors available on that input. Let $\boldsymbol{V}$ be the fixed matrix whose columns are all the unit vectors listed by the span program. On a particular input, the unavailable columns are ignored. On an accepting input (1-input), a 1-witness is a coefficient vector $\boldsymbol{a}$, supported on the available columns, whose linear combination equals the target vector; its size is $\|\boldsymbol{a}\|^2$. On a rejecting input (0-input), a 0-witness is a vector $\boldsymbol{w}$ that is orthogonal to every available vector and has inner product 1 with the target vector; its size is $\|\boldsymbol{w}\|^2$. The balanced witness bound S is a uniform upper bound on both witness sizes over all accepting and rejecting inputs. Our construction will guarantee

$$S = O(\sqrt{n}) \qquad \text{and} \qquad \|\mathrm{abs}(\boldsymbol{V})\| = O(1),$$

where S is the balanced witness bound and $\mathrm{abs}(\boldsymbol{V})$ is obtained from $\boldsymbol{V}$ by taking entrywise absolute values. The norm hypothesis $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$ is about the full fixed matrix $\boldsymbol{V}$, not an input-specific submatrix.

2.1 From Span Programs to Polynomials

In Section 4 we prove the general extraction lemma. If a vector-form span program has balanced witness bound S and $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$, then the computed function has a degree-$O(S)$ approximating polynomial with coefficient ℓ_1-norm $2^{O(S)}$.

For this overview, the important point is the coefficient accounting. The extraction lemma constructs a symmetric symbolic matrix $\widetilde{\boldsymbol{M}}(x)$ whose entries are constants or degree-one polynomials in the input bits. It then defines the approximant to be the scalar $\boldsymbol{o}^\top \Psi(\widetilde{\boldsymbol{M}}(x))\boldsymbol{o}$, where Ψ is a degree-$O(S)$ Chebyshev spectral filter. Expanding Ψ reduces the coefficient accounting to bounding powers of this symbolic matrix. The condition $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$ ensures that these matrix powers do not cause uncontrolled growth in the coefficient ℓ_1-norm. The only exponential loss comes from the coefficient norm of the degree-$O(S)$ filter, giving the final $2^{O(S)}$ coefficient bound.

Thus, the main task is to construct, for every size-n formula, a span program with $S = O(\sqrt{n})$ and $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$.

2.2 Local Gate Composition

We construct the span program for the entire formula from the bottom up. When two child span programs are combined at a parent gate, all their listed unit vectors are retained, with the same availability labels, in the parent program. The composition step defines a new unit target for the parent; this target is not itself an available vector. At an OR gate, the step also adds one always-available unit auxiliary vector.

Here A denotes a worst-case upper bound on the squared norm of a positive witness (a 1-witness), and B denotes the corresponding upper bound for a negative witness (a 0-witness). The witness parameters are balanced when $A = B = S$. Suppose the two child targets are unit vectors

$\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, lying in orthogonal subspaces, with witness parameters (A_1, B_1) and (A_2, B_2). The parent target is a unit vector of the form

$$\boldsymbol{u} := \cos\theta\, \boldsymbol{u}_1 + \sin\theta\, \boldsymbol{u}_2,$$

for a suitable angle θ. For an OR gate, we add the always-available unit vector orthogonal to $\boldsymbol{u}$ in $\operatorname{span}\{\boldsymbol{u}_1, \boldsymbol{u}_2\}$, namely

$$\boldsymbol{r} := \sin\theta\, \boldsymbol{u}_1 - \cos\theta\, \boldsymbol{u}_2.$$

For an AND gate, no such vector is added.

The intuition is simple. At an OR gate, the additional unit vector allows the parent target to be expressed using either child target: if the left child accepts, then $\boldsymbol{u}$ lies in the span of $\boldsymbol{u}_1$ and $\boldsymbol{r}$, and the same holds symmetrically for the right child. At an AND gate, both children must accept, so a positive witness for the parent target combines positive witnesses for both child targets.

The angle θ can be chosen so that the parent witness parameters (A, B) satisfy

$$(A+1)B = (A_1+1)B_1 + (A_2+1)B_2.$$

This identity is proved in Lemma 5.3. In particular, if a terminal target has balanced witness bound S_i, then its contribution to the invariant is $(S_i+1)S_i = S_i^2 + S_i$. At the root, the accumulated invariant is of order n; after the helper-balancing step described below, it yields a balanced witness bound of order $\sqrt{n}$.

Local gate composition alone yields the witness-size recurrence, but it does not by itself bound $\|\operatorname{abs}(\boldsymbol{V})\|$. The problem is that targets for large subformulas can become dense linear combinations of many lower-level coordinates. In a balanced tree, for instance, an auxiliary OR-gate vector near the root resembles a normalized difference between sums over the two large child subtrees. After taking entrywise absolute values, the cancellations inside these dense columns disappear. Many such columns can have positive mass on the same leaf coordinates, so the rows corresponding to those coordinates accumulate contributions from many ancestors. The helper-target construction below prevents this accumulation by keeping dense vectors local to one chain.

2.3 Heavy Chains and Helper Targets

The previous paragraph suggests that we should sometimes prevent a dense target vector from being used higher in the tree. We do so by introducing helper targets at selected nodes. Their placement is governed by a heavy-chain decomposition: at every internal node, choose a child with at least as many descendant leaves as the other child as the heavy child, and call the other child light. The heavy edges form disjoint chains, and we introduce one helper target for each heavy chain.

For a heavy chain C, the terminal targets are the boundary targets that feed into the chain: one comes from each light child hanging off the chain, and one is the leaf at the bottom of the heavy path. Thus each terminal target is either a helper coordinate $\boldsymbol{h}_D$ exported by a light child chain D or a leaf coordinate. All terminal targets are unit vectors. The span program built for C computes the subformula rooted at the top node of C. Its exported target is a fresh unit coordinate $\boldsymbol{h}_C$, and its balanced witness bound is denoted by S_C.

Inside the chain, we first build the natural internal target $\boldsymbol{u}_C$ for the chain formula, using only the terminal targets entering the chain. If these terminal targets have balanced witness bounds $S_1, \ldots, S_k$, then the local gate invariant gives witness parameters (A_C, B_C) satisfying

$$(A_C+1)B_C = \sum_{i=1}^{k} (S_i^2 + S_i).$$

Next, we introduce the fresh helper coordinate $\boldsymbol{h}_C$, make it the target exported by the chain, and add an always-available unit connector vector in the plane spanned by $\boldsymbol{h}_C$ and $\boldsymbol{u}_C$:

$$\boldsymbol{g}_C := \cos\delta_C\, \boldsymbol{h}_C - \sin\delta_C\, \boldsymbol{u}_C.$$

The helper coordinate $\boldsymbol{h}_C$ is the new target exported by the chain; it is not itself an available vector. The connector $\boldsymbol{g}_C$ is always available and is unit. It makes $\boldsymbol{h}_C$ accepted exactly when $\boldsymbol{u}_C$ is accepted. We choose the angle δ_C so that the positive and negative witness sizes for $\boldsymbol{h}_C$ are balanced. This gives

$$S_C = 1 + \sqrt{\sum_{i=1}^{k}(S_i^2 + S_i)}.$$

The parent chain uses only the fresh coordinate $\boldsymbol{h}_C$ as a terminal target, rather than the potentially dense vector $\boldsymbol{u}_C$. Thus dense internal targets are not repeatedly reused by higher chains.

It is useful to explain why long chains are safe. If the entire formula were a single chain and we formally included the orthogonal companion at every gate, the resulting auxiliary gate vectors would form an orthonormal system before entrywise absolute values were taken; the actual OR-gate vectors form a subfamily of this system. After taking entrywise absolute values, the corresponding matrix differs from the orthonormal-system matrix by a simple diagonal-type perturbation. Thus a long chain alone does not create a large entrywise-absolute norm. The difficulty arises when dense targets are repeatedly fed into other dense targets across different branches. The heavy-chain decomposition preserves the safe behavior within each chain while introducing helpers only at the interfaces between chains.

Witness size. The recurrence

$$S_C = 1 + \sqrt{\sum_{i=1}^{k}(S_i^2 + S_i)}$$

combined with the heavy-chain property gives $S_{\mathrm{root}} = O(\sqrt{n})$. The only structural fact needed is that whenever a terminal comes from a light child, the corresponding child subtree contains at most half as many leaves as the subtree rooted at the top of the current chain. In Lemma 5.7, we make this precise by proving inductively that

$$S_C \le 100\sqrt{n_C} - 5,$$

where n_C is the number of leaves in the subtree rooted at the top of the chain C.

2.4 The Entrywise-Absolute Matrix Norm

It remains to prove $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$. The block associated with a chain C is the submatrix consisting of the available unit vectors introduced within C: the auxiliary vectors at OR gates on the chain and the connector vector $\boldsymbol{g}_C$. For the norm analysis, we may also introduce the corresponding formal orthogonal companion vectors at AND gates; the actual OR-gate vectors form a submatrix of this formal orthonormal system.

The signed formal gate vectors within a chain are orthonormal. Moreover, taking entrywise absolute values changes the corresponding matrix by only a constant-norm perturbation, and the helper connector contributes one additional unit vector. Hence each chain block has entrywise-absolute operator norm $O(1)$.

The full span program is obtained by assembling these chain blocks. The rigorous proof, given in Lemma 5.9, uses a bounded-overlap argument. Every helper coordinate appears in its own connector block and in the parent-chain block in which it is used as a terminal target. Every leaf coordinate appears in its unit literal vector and in the unique chain in which it is a terminal target. Therefore, when the block matrices are applied to their coefficient vectors, only $O(1)$ output vectors overlap at any coordinate, so their squared norms sum to at most a constant factor more than they would under orthogonal support. This gives

$$\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$$

for the full span program.

Applying the extraction lemma to the root helper target, together with the witness bound $S = O(\sqrt{n})$ and the global matrix bound $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$, yields a pointwise 1/3-approximating polynomial of degree $O(\sqrt{n})$ and coefficient ℓ_1-norm $2^{O(\sqrt{n})}$.

3 Preliminaries

3.1 Notation

All vector spaces are real Euclidean spaces. Vectors are written in bold lowercase, for instance $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\tau}$. Matrices are written in bold uppercase, for instance $\boldsymbol{M}, \boldsymbol{V}, \boldsymbol{B}$. If $\boldsymbol{M}$ is a real matrix, then $\mathrm{abs}(\boldsymbol{M})$ denotes the matrix obtained by replacing every entry of $\boldsymbol{M}$ by its absolute value. We write $\|\boldsymbol{M}\|$ for its $\ell_2 \to \ell_2$ operator norm. Explicitly, if the input and output spaces are equipped with the usual Euclidean norm, then

$$\|\boldsymbol{M}\| := \sup_{\boldsymbol{x} \neq 0} \frac{\|\boldsymbol{M}\boldsymbol{x}\|_2}{\|\boldsymbol{x}\|_2}.$$

Equivalently, $\|\boldsymbol{M}\|$ is the smallest number L such that $\|\boldsymbol{M}\boldsymbol{x}\|_2 \leq L\,\|\boldsymbol{x}\|_2$ for every vector $\boldsymbol{x}$. We write $U \oplus^{\perp} W$ for the orthogonal direct sum of two Euclidean subspaces, and similarly for more than two summands.

The coefficient ℓ_1-norm of a real polynomial p in n variables, written in the $\{0,1\}$-monomial basis, is denoted

$$\|p\|_1 := \sum_{T \subseteq [n]} |p_T|,$$

where $p(x) = \sum_{T \subseteq [n]} p_T \prod_{i \in T} x_i$ after multilinearization on the Boolean cube.

A De Morgan formula means a binary formula over $\{\wedge, \vee, \neg\}$, with negations pushed to the leaves; XOR gates are not part of the basis. The size of a formula is the number of leaf occurrences. Repeated occurrences of the same input variable are counted with multiplicity.

3.2 Vector-Form Span Programs

We use the following vector-form version of span programs. The term “vector-form” serves only to distinguish this representation from matrix-kernel formulations: the span program is specified directly by a unit target vector and by unit vectors that may or may not be available on a given input.

Definition 3.1 (Vector-form span program)**.** *A* vector-form span program *consists of a real Euclidean space H, a unit target vector $\boldsymbol{\tau} \in H$, and a finite list of unit vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m \in H$. Each listed vector has one of the following labels:* always available, *x_i, or $\neg x_i$.*

On input x, a vector labeled always available is available; a vector labeled x_i is available exactly when $x_i = 1$; and a vector labeled $\neg x_i$ is available exactly when $x_i = 0$. Let $I_x \subseteq \{1, \dots, m\}$ be the set of available indices. The span program accepts x iff

$$\boldsymbol{\tau} \in \operatorname{span}\{\boldsymbol{v}_j : j \in I_x\}.$$

Let $\boldsymbol{V}$ be the matrix with columns $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$. For input x, let $\boldsymbol{V}_x$ be the matrix whose j-th column is $\boldsymbol{v}_j$ if $j \in I_x$, and is zero otherwise.

- *On an accepting input, a* 1-witness *is a coefficient vector $\boldsymbol{a}$ such that $\boldsymbol{V}_x \boldsymbol{a} = \boldsymbol{\tau}$. Its size is defined by $\|\boldsymbol{a}\|^2$.*
- *On a rejecting input, a* 0-witness *is a vector $\boldsymbol{w} \in H$ such that $\boldsymbol{w}^\top \boldsymbol{\tau} = 1$ and $\boldsymbol{w}^\top \boldsymbol{V}_x = 0$. Its size is defined by $\|\boldsymbol{w}\|^2$.*

Thus every vector available on any input is unit. Throughout the paper, every intermediate or exported target vector introduced by the construction is also unit.

Definition 3.2 (Witness parameters and balanced witness bound)**.** *Let $\boldsymbol{\tau}$ be a unit target. For $A, B \geq 1$, it has* witness parameters (A, B) *if, on every 1-input, there is a 1-witness of size at most A, and on every 0-input, there is a 0-witness of size at most B. It has* balanced witness bound *$S \geq 1$ if both witness sizes are at most S; equivalently, it has witness parameters (S, S). All witness sizes are squared Euclidean norms.*

4 Constructing Polynomials from Span Programs

In this section we show how to turn a span program with small balanced witness bound into a low-degree approximating polynomial. The conversion also needs the additional hypothesis that the entrywise-absolute matrix of available vectors has bounded operator norm; this is the condition that will later give the coefficient ℓ_1-norm bound. In Section 5 we construct, for every De Morgan formula, a span program satisfying these hypotheses.

The following lemma is the span-program to polynomial step. This argument is standard in the span-program approach to formula evaluation, but we give the proof because we need the coefficient ℓ_1-norm bound.

Lemma 4.1 (Constructing a polynomial from a span program)**.** *Let a vector-form span program compute a Boolean function f, and suppose it has balanced witness bound at most S, where $S \geq 1$. Let $\boldsymbol{V}$ be the matrix from Definition 3.1, whose columns are the span-program vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$. If*

$$\|\operatorname{abs}(\boldsymbol{V})\| = O(1),$$

then f has a real polynomial P satisfying

$$|P(x) - f(x)| \leq \frac{1}{3}$$

for all Boolean inputs x, with

$$\deg P = O(S), \qquad \|P\|_1 \leq 2^{O(S)}.$$

Before the proof, let us indicate the role of the matrix $\boldsymbol{M}_x$ and vector $\boldsymbol{o}$ below. It is designed so that on accepting inputs, $\boldsymbol{o}$ has large projection onto $\boldsymbol{M}_x$'s zero-eigenspace, while on rejecting inputs it has very small projection onto the $O(1/S)$-small eigenspace. A Chebyshev polynomial then filters the spectrum. The bounded entrywise-absolute norm hypothesis is used only in the final coefficient-ℓ_1-norm accounting.

Proof. The proof has two logically separate parts. First we use the span program to build a matrix whose small-eigenvalue behavior distinguishes accepting inputs from rejecting inputs. Then we apply a polynomial spectral filter to this matrix and bound the coefficient ℓ_1-norm of the resulting polynomial.

The Matrix.

Let $\boldsymbol{o}$ be a new unit coordinate. The matrix below acts on

$$\mathbb{R}\boldsymbol{o} \oplus^{\perp} H \oplus^{\perp} \mathbb{R}^m,$$

where $\mathbb{R}\boldsymbol{o} := \{r\boldsymbol{o} \colon r \in \mathbb{R}\}$, H is the span-program ambient space, and $\mathbb{R}^m$ indexes the available vectors. Here $\oplus^{\perp}$ denotes an orthogonal direct sum. For a Boolean input x, let $\boldsymbol{V}_x \colon \mathbb{R}^m \to H$ be the matrix from Definition 3.1. Define the symmetric matrix

$$\boldsymbol{M}_x := \begin{pmatrix} 0 & \frac{1}{10\sqrt{S}}\boldsymbol{\tau}^{\top} & 0 \\ \frac{1}{10\sqrt{S}}\boldsymbol{\tau} & 0 & \boldsymbol{V}_x \\ 0 & \boldsymbol{V}_x^{\top} & 0 \end{pmatrix}.$$

Since $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$ and $\|\boldsymbol{\tau}\| = 1$, we have $\|\mathrm{abs}(\boldsymbol{M}_x)\| = O(1)$, uniformly over inputs x. Choose a sufficiently large absolute constant c_{norm} that is independent of x, and set

$$\widetilde{\boldsymbol{M}}_x := \frac{1}{c_{\mathrm{norm}}}\boldsymbol{M}_x$$

so that $\left\|\mathrm{abs}(\widetilde{\boldsymbol{M}}_x)\right\| \le 0.1$. In particular, all eigenvalues of $\widetilde{\boldsymbol{M}}_x$ lie in $[-1, 1]$.

Eigenvalue Properties on Accepting Inputs.

Consider an input x such that $f(x) = 1$. Let $\boldsymbol{a} \in \mathbb{R}^m$ be a 1-witness, so $\boldsymbol{V}_x\boldsymbol{a} = \boldsymbol{\tau}$ and $\|\boldsymbol{a}\|^2 \le S$. Embed $\boldsymbol{a}$ in the vector-index block of $\mathbb{R}\boldsymbol{o} \oplus^{\perp} H \oplus^{\perp} \mathbb{R}^m$. Define

$$\boldsymbol{\psi} := \begin{pmatrix} 1 \\ \boldsymbol{0}_H \\ -\frac{1}{10\sqrt{S}}\boldsymbol{a} \end{pmatrix} = \boldsymbol{o} - \frac{1}{10\sqrt{S}}\boldsymbol{a}.$$

Then $\boldsymbol{M}_x\boldsymbol{\psi} = 0$, hence also $\widetilde{\boldsymbol{M}}_x\boldsymbol{\psi} = 0$. Moreover,

$$\|\boldsymbol{\psi}\|^2 = 1 + \frac{\|\boldsymbol{a}\|^2}{100S} \le 1 + \frac{1}{100}.$$

Also $\boldsymbol{o}^{\top}\boldsymbol{\psi} = 1$. Since $\boldsymbol{\psi}$ lies in the zero-eigenspace of $\widetilde{\boldsymbol{M}}_x$, the squared projection of $\boldsymbol{o}$ onto that zero-eigenspace is at least

$$\frac{|\boldsymbol{o}^{\top}\boldsymbol{\psi}|^2}{\|\boldsymbol{\psi}\|^2} \ge \frac{1}{1 + 1/100} > 0.99.$$

Eigenvalue Properties on Rejecting Inputs.

Suppose $f(x) = 0$. Let $\boldsymbol{w} \in H$ be a 0-witness:

$$\boldsymbol{w}^\top \boldsymbol{\tau} = 1, \qquad \boldsymbol{w}^\top \boldsymbol{V}_x = 0, \qquad \|\boldsymbol{w}\|^2 \leq S.$$

We will show that any vector in the small-eigenvalue subspace of $\widetilde{\boldsymbol{M}}_x$ has very small component in the $\boldsymbol{o}$ coordinate. Let $\varepsilon > 0$, to be chosen later, and let $\boldsymbol{\phi}$ lie in the spectral subspace of $\widetilde{\boldsymbol{M}}_x$ with eigenvalues of absolute value at most ε. Then

$$\left\| \widetilde{\boldsymbol{M}}_x \boldsymbol{\phi} \right\| \leq \varepsilon \left\| \boldsymbol{\phi} \right\|,$$

and therefore

$$\left\| \boldsymbol{M}_x \boldsymbol{\phi} \right\| \leq c_{\mathrm{norm}} \varepsilon \left\| \boldsymbol{\phi} \right\|.$$

Write

$$\boldsymbol{\phi} = \begin{pmatrix} \phi_o \\ \boldsymbol{\phi}_H \\ \boldsymbol{\phi}_J \end{pmatrix},$$

where $\boldsymbol{\phi}_H \in H$ and $\boldsymbol{\phi}_J \in \mathbb{R}^m$. The H-block of $\boldsymbol{M}_x \boldsymbol{\phi}$ is

$$\frac{\phi_o}{10\sqrt{S}} \boldsymbol{\tau} + \boldsymbol{V}_x \boldsymbol{\phi}_J.$$

We view $\boldsymbol{w}$ as a vector in the middle H-block, extended by zero on the other two blocks. Multiplying this H-block on the left by $\boldsymbol{w}^\top$, and using $\boldsymbol{w}^\top \boldsymbol{V}_x = 0$, gives

$$\boldsymbol{w}^\top \boldsymbol{M}_x \boldsymbol{\phi} = \boldsymbol{w}^\top \left(\frac{\phi_o}{10\sqrt{S}} \boldsymbol{\tau} + \boldsymbol{V}_x \boldsymbol{\phi}_J \right) = \frac{\phi_o}{10\sqrt{S}}.$$

Hence

$$\frac{|\phi_o|}{10\sqrt{S}} \leq \|\boldsymbol{w}\| \left\| \boldsymbol{M}_x \boldsymbol{\phi} \right\| \leq c_{\mathrm{norm}} \sqrt{S}\, \varepsilon \left\| \boldsymbol{\phi} \right\|.$$

Choose

$$\varepsilon := \frac{1}{1000 c_{\mathrm{norm}} S} \in \Theta(1/S).$$

Then

$$|\phi_o| \leq 0.01 \left\| \boldsymbol{\phi} \right\|. \tag{1}$$

Let $\boldsymbol{P}_{\leq \varepsilon}$ be the orthogonal projector onto the eigenspaces of $\widetilde{\boldsymbol{M}}_x$ whose eigenvalues λ satisfy $|\lambda| \leq \varepsilon$. Applying Equation (1) to $\boldsymbol{\phi} := \boldsymbol{P}_{\leq \varepsilon} \boldsymbol{o}$, and using that $\boldsymbol{P}_{\leq \varepsilon}$ is an orthogonal projector, we have

$$\|\boldsymbol{P}_{\leq \varepsilon} \boldsymbol{o}\|^2 = \boldsymbol{o}^\top \left(\boldsymbol{P}_{\leq \varepsilon} \boldsymbol{o} \right) \leq 0.01 \|\boldsymbol{P}_{\leq \varepsilon} \boldsymbol{o}\|,$$

and therefore

$$\|\boldsymbol{P}_{\leq \varepsilon} \boldsymbol{o}\| \leq 0.01.$$

A Chebyshev Polynomial Filter.

We now choose a univariate polynomial that distinguishes the spectral behavior just proved: the polynomial is large at eigenvalue zero and small outside the interval $(-\varepsilon, \varepsilon)$. We obtain this by first constructing a bounded Chebyshev filter Φ, and then setting $\Psi = \Phi^2$ to make the final filter nonnegative.

Let T_d be the Chebyshev polynomial of the first kind, characterized by $T_d(\cos\alpha) = \cos(d\alpha)$. Equivalently, $T_d\left((r+r^{-1})/2\right) = (r^d + r^{-d})/2$. Choose $d = \lceil c_{\mathrm{cheb}}/\varepsilon \rceil$, where the absolute constant c_{cheb} will be chosen large enough. Since $\varepsilon = \Theta(1/S)$, this gives $d = \Theta(1/\varepsilon) = \Theta(S)$. Define

$$\Phi(t) := \frac{T_d\left(1 + \frac{2(\varepsilon^2 - t^2)}{1-\varepsilon^2}\right)}{T_d\left(1 + \frac{2\varepsilon^2}{1-\varepsilon^2}\right)}.$$

The polynomial Φ has the following properties:

- At $t = 0$, the numerator and denominator are equal, so $\Phi(0) = 1$.
- If $|t| \in [\varepsilon, 1]$, then
$$1 + \frac{2(\varepsilon^2 - t^2)}{1 - \varepsilon^2} \in [-1, 1].$$
Hence the numerator has absolute value at most 1. The denominator is $T_d\left(1 + \frac{2\varepsilon^2}{1-\varepsilon^2}\right)$. We use the standard estimate
$$T_d(1+\rho) \geq \frac{1}{2}\exp(\Omega(d\sqrt{\rho})) \qquad (0 < \rho \leq 1)$$
which follows from the identity $T_d((r+r^{-1})/2) = (r^d + r^{-d})/2$. Here $\rho = 2\varepsilon^2/(1-\varepsilon^2) = \Theta(\varepsilon^2)$. Thus choosing c_{cheb} sufficiently large makes the denominator at least 10. Therefore $|\Phi(t)| \leq 1/10$ for $|t| \in [\varepsilon, 1]$.
- If $|t| \leq \varepsilon$, then
$$1 \leq 1 + \frac{2(\varepsilon^2 - t^2)}{1-\varepsilon^2} \leq 1 + \frac{2\varepsilon^2}{1-\varepsilon^2}.$$
Since T_d is increasing on $[1, \infty)$, we have $|\Phi(t)| \leq 1$ for $|t| \leq \varepsilon$.

Thus $|\Phi(t)| \leq 1$ for all $|t| \leq 1$. Set
$$\Psi(t) := \Phi(t)^2.$$
Then
$$\Psi(0) = 1, \qquad 0 \leq \Psi(t) \leq 1 \quad (|t| \leq 1),$$
and
$$\Psi(t) \leq \frac{1}{100} \qquad (|t| \in [\varepsilon, 1]).$$
Moreover, $\deg \Psi = O(S)$.

Now we present the construction of the final polynomial. We first define the symbolic matrix-valued polynomial. Let $\boldsymbol{V}(x)$ be obtained from $\boldsymbol{V}$ by leaving always-available columns unchanged, multiplying a column labeled x_i by the variable x_i, and multiplying a column labeled $\neg x_i$ by $1 - x_i$. Define
$$\boldsymbol{M}(x) := \begin{pmatrix} 0 & \frac{1}{10\sqrt{S}}\boldsymbol{\tau}^\top & 0 \\ \frac{1}{10\sqrt{S}}\boldsymbol{\tau} & 0 & \boldsymbol{V}(x) \\ 0 & \boldsymbol{V}(x)^\top & 0 \end{pmatrix}, \qquad \widetilde{\boldsymbol{M}}(x) := \frac{1}{c_{\mathrm{norm}}}\boldsymbol{M}(x).$$

For every Boolean input x, $\widetilde{\boldsymbol{M}}(x)$ evaluates to the numeric matrix $\widetilde{\boldsymbol{M}}_x$ used above. We now define the approximating polynomial by

$$P(x) := \boldsymbol{o}^\top \Psi(\widetilde{\boldsymbol{M}}(x))\boldsymbol{o}.$$

We now verify pointwise approximation using the fixed-input matrix $\widetilde{\boldsymbol{M}}_x = \widetilde{\boldsymbol{M}}(x)$. Recall that its eigenvalues lie in $[-1, 1]$ by the choice of c_{norm}.

- If $f(x) = 1$, write the orthogonal decomposition $\boldsymbol{o} = \boldsymbol{o}_0 + \boldsymbol{o}_\perp$, where $\boldsymbol{o}_0$ is the projection onto the zero-eigenspace of $\widetilde{\boldsymbol{M}}_x$. We proved $\|\boldsymbol{o}_0\|^2 \geq 0.99$. Since $\Psi(0) = 1$ and $\Psi \geq 0$ on $[-1, 1]$,

  $$P(x) = \boldsymbol{o}^\top \Psi(\widetilde{\boldsymbol{M}}_x)\boldsymbol{o} \geq \|\boldsymbol{o}_0\|^2 \geq 0.99.$$

  Also $P(x) \leq 1$, because $0 \leq \Psi \leq 1$ on $[-1, 1]$.

- If $f(x) = 0$, decompose $\boldsymbol{o} = \boldsymbol{o}_{\leq\varepsilon} + \boldsymbol{o}_{>\varepsilon}$, where $\boldsymbol{o}_{\leq\varepsilon} = \boldsymbol{P}_{\leq\varepsilon}\boldsymbol{o}$. We proved $\|\boldsymbol{o}_{\leq\varepsilon}\| \leq 0.01$, so its squared norm is at most 10^{-4}. On the remaining spectral subspace all eigenvalues satisfy $|\lambda| \in [\varepsilon, 1]$, and thus $\Psi(\lambda) \leq 1/100$. Hence

  $$0 \leq P(x) \leq \|\boldsymbol{o}_{\leq\varepsilon}\|^2 + \frac{1}{100}\|\boldsymbol{o}_{>\varepsilon}\|^2 \leq 10^{-4} + \frac{1}{100} < 0.02.$$

Hence P pointwise 1/3-approximates f.

Degree and Coefficient ℓ_1-Norm.

The preceding argument fixed an input x and treated $\widetilde{\boldsymbol{M}}_x$ as a numeric matrix. For degree and coefficient ℓ_1-norm bounds, we use the symbolic matrix $\widetilde{\boldsymbol{M}}(x)$ defined above.

Since each entry of $\widetilde{\boldsymbol{M}}(x)$ is a degree-1 polynomial in the input bits and $\deg \Psi = O(S)$, we have $\deg P = O(S)$.

In the rest of the proof, coefficient norms are taken before Boolean multilinearization. At the end we multilinearize on the Boolean cube; this only merges coefficients of monomials and therefore cannot increase coefficient ℓ_1-norm.

It remains to bound the coefficient ℓ_1-norm. For a matrix $\boldsymbol{Q}(x)$ whose entries are polynomials, define $\mathcal{L}(\boldsymbol{Q})$ entrywise by

$$\mathcal{L}(\boldsymbol{Q})_{ij} := \|Q_{ij}\|_1.$$

In words, $\mathcal{L}(\boldsymbol{Q})$ records the coefficient ℓ_1-norm of each polynomial entry of $\boldsymbol{Q}$. Coefficient ℓ_1-norm is submultiplicative, so

$$\mathcal{L}(\boldsymbol{QR}) \leq \mathcal{L}(\boldsymbol{Q})\mathcal{L}(\boldsymbol{R}) \qquad \text{entrywise.}$$

For the symbolic matrix $\widetilde{\boldsymbol{M}}(x)$, define the fixed nonnegative majorant

$$\boldsymbol{L}_{\text{maj}} := \frac{1}{c_{\text{norm}}} \begin{pmatrix} 0 & \frac{1}{10\sqrt{S}}\operatorname{abs}(\boldsymbol{\tau})^\top & 0 \\ \frac{1}{10\sqrt{S}}\operatorname{abs}(\boldsymbol{\tau}) & 0 & 2\operatorname{abs}(\boldsymbol{V}) \\ 0 & 2\operatorname{abs}(\boldsymbol{V})^\top & 0 \end{pmatrix}.$$

This matrix is not evaluated on an input; it only records upper bounds on coefficient ℓ_1-norms of the entries of $\widetilde{\boldsymbol{M}}(x)$. The factor 2 appears because $\|x_i\|_1 = 1$ while $\|1 - x_i\|_1 = 2$. Thus

$$0 \leq \mathcal{L}(\widetilde{\boldsymbol{M}}) \leq \boldsymbol{L}_{\text{maj}} \qquad \text{entrywise.}$$

We use the elementary monotonicity fact that if $0 \le \boldsymbol{A} \le \boldsymbol{B}$ entrywise, then $\|\boldsymbol{A}\| \le \|\boldsymbol{B}\|$: for every vector $\boldsymbol{x}$, $|\boldsymbol{A}\boldsymbol{x}| \le \boldsymbol{A}|\boldsymbol{x}| \le \boldsymbol{B}|\boldsymbol{x}|$ entrywise, and hence $\|\boldsymbol{A}\boldsymbol{x}\| \le \|\boldsymbol{B}\|\,\|\boldsymbol{x}\|$. Since $\|\mathrm{abs}(\boldsymbol{V})\| = O(1)$ and $\|\boldsymbol{\tau}\| = 1$, our choice of c_{norm} gives

$$\left\|\mathcal{L}(\widetilde{\boldsymbol{M}})\right\| \le 1.$$

The next two claims isolate the coefficient accounting.

Claim 4.1.1. *For every integer $j \ge 0$,*

$$\left\|\boldsymbol{o}^\top \widetilde{\boldsymbol{M}}(x)^j \boldsymbol{o}\right\|_1 \le 1.$$

Proof. By entrywise submultiplicativity,

$$\mathcal{L}(\widetilde{\boldsymbol{M}}^j) \le \mathcal{L}(\widetilde{\boldsymbol{M}})^j \qquad \text{entrywise.}$$

Therefore

$$\left\|\mathcal{L}(\widetilde{\boldsymbol{M}}^j)\right\| \le \left\|\mathcal{L}(\widetilde{\boldsymbol{M}})\right\|^j \le 1.$$

The matrix $\mathcal{L}(\widetilde{\boldsymbol{M}}^j)$ is nonnegative, so each of its entries is bounded by its operator norm. The entry corresponding to $\boldsymbol{o}^\top \widetilde{\boldsymbol{M}}(x)^j \boldsymbol{o}$ therefore has coefficient ℓ_1-norm at most 1. □

Write

$$\Psi(t) = \sum_{j=0}^{\deg \Psi} \alpha_j t^j.$$

Claim 4.1.2. *We have*

$$\sum_{j=0}^{\deg \Psi} |\alpha_j| \le 2^{O(S)}.$$

Proof. The recurrence $T_{d+1}(t) = 2tT_d(t) - T_{d-1}(t)$ implies that the monomial coefficient ℓ_1-norm of T_d is at most 3^d. Let

$$A_\varepsilon(t) := 1 + \frac{2(\varepsilon^2 - t^2)}{1 - \varepsilon^2}.$$

Since $\|A_\varepsilon\|_1 = O(1)$, coefficient ℓ_1-norm submultiplicativity gives

$$\|T_d(A_\varepsilon(t))\|_1 \le \sum_r |c_r|\, \|A_\varepsilon\|_1^r \le \|T_d\|_1 \max(1, \|A_\varepsilon\|_1)^d \le 2^{O(d)},$$

where $T_d(t) = \sum_r c_r t^r$. The denominator in Φ has absolute value at least 10, and squaring Φ to form Ψ only squares the coefficient ℓ_1-norm. Since $d = O(S)$, the claim follows. □

It follows from Claims 4.1.1 and 4.1.2 that

$$\|P\|_1 \le \sum_{j=0}^{\deg \Psi} |\alpha_j| \left\|\boldsymbol{o}^\top \widetilde{\boldsymbol{M}}(x)^j \boldsymbol{o}\right\|_1 \le 2^{O(S)}.$$

Finally, multilinearizing P on the Boolean cube preserves its values there and can only merge coefficients of monomials, so it does not increase the coefficient ℓ_1-norm. This proves the lemma. □

5 Proof of the Main Theorem

In this section we prove Theorem 1.1, restated below as Theorem 5.1. The proof has two steps. We construct span programs for formula subtrees, maintaining both a balanced witness bound and a uniform bound on the entrywise-absolute norm of the available-vector matrix. We then apply Lemma 4.1 to the span program computing the formula.

Theorem 5.1 (Main theorem, restated from Theorem 1.1)**.** *Let $F : \{0,1\}^m \to \{0,1\}$ be computed by a De Morgan formula with n leaf occurrences. Then there is a real multilinear polynomial P in the m input variables such that, for every $x \in \{0,1\}^m$,*

$$|P(x) - F(x)| \leq \frac{1}{3},$$

and

$$\deg P = O(\sqrt{n}), \qquad \|P\|_1 \leq 2^{O(\sqrt{n})}.$$

For the construction, it is enough to consider a read-once monotone formula in formal leaf-occurrence variables $y_1, \ldots, y_n$. A general De Morgan formula computing a function of variables $x_1, \ldots, x_m$ can first be viewed in these formal variables; at the end each y_ℓ is replaced by the original literal x_i or $1 - x_i$ labeling that leaf occurrence. This final substitution is handled in the completion of the proof.

We prove the theorem by constructing inductively, for each relevant subtree, a vector-form span program computing that subtree. The target of a completed subtree will be a unit vector, and its balanced witness bound will be denoted by an uppercase parameter such as S, with subscripts indicating the corresponding subtree or chain. Leaves are handled first. The gate-composition lemma then explains how to combine two child span programs. Heavy chains and helper targets are used afterward to keep the entrywise-absolute matrix norm bounded.

5.1 Leaf Programs and Gate Composition

Lemma 5.2 (Leaves)**.** *For every formal leaf variable y_ℓ, there is a vector-form span program computing the function y_ℓ, with unit target $\boldsymbol{e}_\ell$, all available vectors unit, and balanced witness bound $S_\ell = 1$.*

Proof. Introduce a fresh one-dimensional coordinate and let $\boldsymbol{e}_\ell$ be its unit basis vector. The target is $\boldsymbol{e}_\ell$. Add one available vector, also equal to $\boldsymbol{e}_\ell$, labeled by the variable y_ℓ. If $y_\ell = 1$, the coefficient 1 on this available vector is a 1-witness of size 1. If $y_\ell = 0$, then $\boldsymbol{w} = \boldsymbol{e}_\ell$ is a 0-witness: it satisfies $\boldsymbol{w}^\top \boldsymbol{e}_\ell = 1$ and is orthogonal to all available vectors, since the only labeled vector is unavailable. Thus both witness sizes are at most 1, so $S_\ell = 1$. □

The next lemma describes how to combine two span programs whose entire ambient spaces are orthogonal. In the formula application, these programs compute the two child subformulas of a gate. The lemma constructs a span program for the parent formula, using all available vectors from the two child span programs and, in the OR case, one additional always-available unit vector.

Lemma 5.3 (Gate composition)**.** *Let P_1 and P_2 be vector-form span programs computing two Boolean functions f_1 and f_2. Let their unit targets be $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$, and assume their ambient spaces are orthogonal. Assume also that all available vectors in the two programs are unit and that their witness parameters are (A_1, B_1) and (A_2, B_2).*

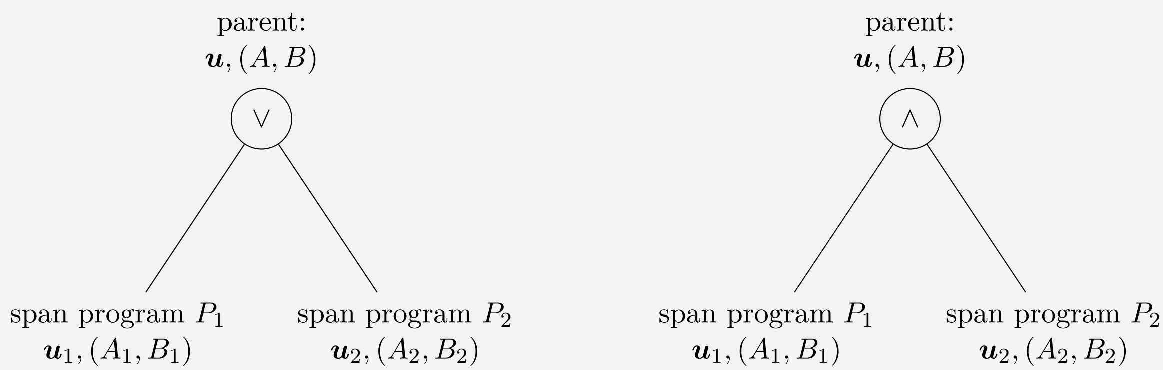

Figure 1: Gate-composition schematic for the OR and AND cases. The parent target $\boldsymbol{u}$ has witness parameters (A, B), while the child targets have parameters (A_1, B_1) and (A_2, B_2).

For a top OR gate, there is a vector-form span program computing $f_1 \vee f_2$. For a top AND gate, there is a vector-form span program computing $f_1 \wedge f_2$. In either case, the parent target is a unit vector $\boldsymbol{u}$ in the direct sum of the two child ambient spaces. The new program includes all available vectors from the child programs; in addition, the OR construction adds one always-available unit vector, while the AND construction adds no new available vector. If (A, B) are the witness parameters of the parent target, then

$$(A + 1)B = (A_1 + 1)B_1 + (A_2 + 1)B_2.$$

Proof. We prove the OR and AND cases separately. In both cases the available vectors from the two child programs are included in the parent program. The new target $\boldsymbol{u}$ is a unit nonnegative linear combination of the two child targets $\boldsymbol{u}_1, \boldsymbol{u}_2$.

OR top gate.

Choose $\theta \in (0, \pi/2)$ such that

$$\cos^2\theta = \frac{A_1 + 1}{A_1 + A_2 + 2}, \qquad \sin^2\theta = \frac{A_2 + 1}{A_1 + A_2 + 2}.$$

Define the unit target

$$\boldsymbol{u} := \cos\theta\, \boldsymbol{u}_1 + \sin\theta\, \boldsymbol{u}_2.$$

This vector is a linear combination of child targets; it is not a new coordinate. Add the always-available unit vector

$$\boldsymbol{r} := \sin\theta\, \boldsymbol{u}_1 - \cos\theta\, \boldsymbol{u}_2.$$

This vector is called the *rotation vector* of this OR gate.

Suppose the OR gate is true. If the left child is true, let $\boldsymbol{a}_1$ be a 1-witness for $\boldsymbol{u}_1$, so $\|\boldsymbol{a}_1\|^2 \leq A_1$. Then

$$\boldsymbol{u} = \frac{1}{\cos\theta}\boldsymbol{u}_1 - \frac{\sin\theta}{\cos\theta}\boldsymbol{r}.$$

Thus a 1-witness for $\boldsymbol{u}$ is obtained by taking $\boldsymbol{a}_1/\cos\theta$ on the columns of P_1, coefficient $-\sin\theta/\cos\theta$ on the always-available vector $\boldsymbol{r}$, and zero on all columns of P_2. Its size is at most

$$\frac{A_1}{\cos^2\theta} + \frac{\sin^2\theta}{\cos^2\theta} = A_1 + A_2 + 1.$$

The right-child case is symmetric. Hence the parent positive witness size is bounded by

$$A := A_1 + A_2 + 1.$$

Now suppose the OR gate is false. Then both children are false. Let $\boldsymbol{w}_1, \boldsymbol{w}_2$ be 0-witnesses for the two child targets, with $\|\boldsymbol{w}_i\|^2 \le B_i$. Since each $\boldsymbol{w}_i$ is orthogonal to every available vector in its child program, and the two child ambient spaces are orthogonal, the vector

$$\boldsymbol{w} := \cos\theta\, \boldsymbol{w}_1 + \sin\theta\, \boldsymbol{w}_2$$

is orthogonal to every available child vector. Moreover,

$$\boldsymbol{w}^\top \boldsymbol{u} = \cos^2\theta\, \boldsymbol{w}_1^\top \boldsymbol{u}_1 + \sin^2\theta\, \boldsymbol{w}_2^\top \boldsymbol{u}_2 = \cos^2\theta + \sin^2\theta = 1,$$

and

$$\boldsymbol{w}^\top \boldsymbol{r} = \cos\theta\sin\theta\, \boldsymbol{w}_1^\top \boldsymbol{u}_1 - \sin\theta\cos\theta\, \boldsymbol{w}_2^\top \boldsymbol{u}_2 = 0.$$

Thus $\boldsymbol{w}$ is a valid 0-witness for the OR target, including the new always-available vector $\boldsymbol{r}$. Its size is at most

$$B := \cos^2\theta B_1 + \sin^2\theta B_2.$$

Therefore

$$\begin{aligned}(A+1)B &= (A_1 + A_2 + 2)\left(\cos^2\theta B_1 + \sin^2\theta B_2\right)\\ &= (A_1 + A_2 + 2)\left(\frac{A_1+1}{A_1+A_2+2}B_1 + \frac{A_2+1}{A_1+A_2+2}B_2\right)\\ &= (A_1+1)B_1 + (A_2+1)B_2.\end{aligned}$$

AND top gate.

Choose $\theta \in (0, \pi/2)$ such that[4]

$$\cos^2\theta = \frac{B_1}{B_1+B_2}, \qquad \sin^2\theta = \frac{B_2}{B_1+B_2}.$$

Define the unit target

$$\boldsymbol{u} := \cos\theta\, \boldsymbol{u}_1 + \sin\theta\, \boldsymbol{u}_2.$$

No new available vector is added.

If the AND gate is true, both children are true. Let $\boldsymbol{a}_1, \boldsymbol{a}_2$ be 1-witnesses for $\boldsymbol{u}_1, \boldsymbol{u}_2$, respectively. Then the parent 1-witness uses coefficients $\cos\theta\, \boldsymbol{a}_1$ on the first child and $\sin\theta\, \boldsymbol{a}_2$ on the second child. Its size is at most

$$A := \cos^2\theta A_1 + \sin^2\theta A_2.$$

If the AND gate is false, at least one child is false. If the left child is false, let $\boldsymbol{w}_1$ be a 0-witness for $\boldsymbol{u}_1$. Then $\boldsymbol{w}_1/\cos\theta$ is a 0-witness for $\boldsymbol{u}$. It has inner product 1 with $\boldsymbol{u}$, is orthogonal to all available vectors from the left child by definition, and is orthogonal to all vectors from the right child because the two child ambient spaces are orthogonal. Its size is at most

$$\frac{B_1}{\cos^2\theta} = B_1 + B_2.$$

[4]By Definition 3.2, the witness parameters satisfy $B_1, B_2 \ge 1$. Therefore both fractions below lie strictly between 0 and 1, so this choice of θ is nondegenerate.

The right-child case is symmetric.

Now

$$(A+1)B = \left(\cos^2\theta A_1 + \sin^2\theta A_2 + 1\right)(B_1+B_2)$$
$$= \left(\frac{B_1}{B_1+B_2}A_1 + \frac{B_2}{B_1+B_2}A_2 + 1\right)(B_1+B_2)$$
$$= (A_1+1)B_1 + (A_2+1)B_2.$$

This proves the lemma. □

5.2 Heavy Chains and Helper Targets

We next organize the bottom-up construction by a heavy-light decomposition. At every internal node, choose the child with more descendant leaves as the heavy child; in case of a tie, choose either child arbitrarily. Call the other child light. The heavy edges form disjoint directed paths.

Definition 5.4 (Heavy chains and terminal targets)**.** A heavy chain is a maximal path that starts at the root or at an internal light child, follows heavy edges, and ends at a leaf. If a light child is itself a leaf, it contributes a leaf terminal rather than starting a new chain. Concretely, a chain C consists of internal nodes $v_1, \ldots, v_{k-1}$ connected by heavy edges, followed by a final leaf on the heavy path. For each internal node v_j, the child not lying on the heavy path contributes one terminal target $\boldsymbol{z}_j$. The last terminal target $\boldsymbol{z}_k$ is the leaf coordinate at the bottom of the heavy path. Thus the terminal targets of C are the helper targets exported by internal light children, the leaf targets contributed by light children that are leaves, and the bottom leaf target.

For a chain C, we use n_C for the number of leaves below the top node of C.

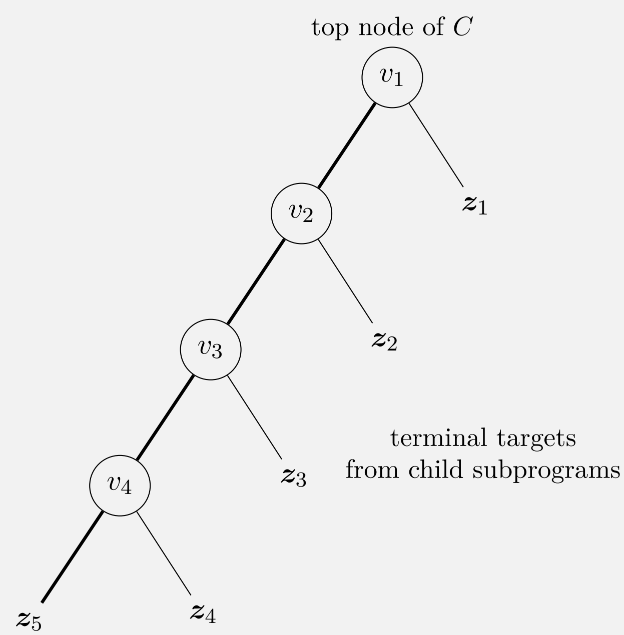

Figure 2: A heavy chain C. The thick path is the heavy path, the side targets $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_4$ come from light-child subprograms, and $\boldsymbol{z}_5$ is the bottom leaf target.

One could apply Lemma 5.3 directly throughout the whole formula tree. This would compute the formula, but the targets of large subformulas could be dense vectors. After taking entrywise absolute values, the cancellations inside these dense columns disappear; many such columns can have positive mass on the same leaf coordinates. This can make the entrywise-absolute matrix norm grow, which would break the hypothesis required in Lemma 4.1. To avoid this, every completed heavy chain C is summarized by a fresh helper coordinate $\boldsymbol{h}_C$. This helper coordinate is a new orthogonal dimension and becomes the target of the span program for the subtree rooted at the top node of C. A parent chain therefore sees $\boldsymbol{h}_C$, a unit coordinate, rather than a dense vector for the whole light subtree.

The inductive object associated with a completed heavy chain C is a span program computing the formula subtree rooted at the top node of C. Its target is $\boldsymbol{h}_C$. The program contains the available vectors introduced inside the chain and from its light-child subprograms. The coordinate $\boldsymbol{h}_C$ may later be used as a terminal target in the parent chain, but $\boldsymbol{h}_C$ itself is not an available vector. All helper coordinates are fresh orthogonal dimensions. The internal targets $\boldsymbol{u}_C$ constructed below are not new dimensions; they are linear combinations of previously constructed terminal targets.

We maintain that all available vectors are unit, and that the target $\boldsymbol{h}_C$ has balanced witness bound S_C.

For every heavy chain C, introduce a fresh unit coordinate

$$\boldsymbol{h}_C.$$

This is a genuine new ambient dimension, and it will be the actual target of the span program for the subtree rooted at the top node of C. Thus the outgoing target of C is $\boldsymbol{h}_C$.

For this same chain C, each terminal target $\boldsymbol{z}_i$ is either the target $\boldsymbol{e}_\ell$ of a leaf program, or the helper target $\boldsymbol{h}_D$ of a light child chain D. These vectors are mutually orthogonal unit coordinates. Let S_i be the already constructed balanced witness bound of $\boldsymbol{z}_i$. For a leaf, $S_i = 1$. For a light child helper $\boldsymbol{h}_D$, $S_i = S_D$. Because each $\boldsymbol{z}_i$ is a unit target, multiplying a witness contribution by a scalar γ multiplies its squared witness cost by γ^2.

In Lemma 5.5, we first ignore the outgoing helper coordinate and apply Lemma 5.3 directly along the chain to build the natural target $\boldsymbol{u}_C$ for the formula computed by the chain. This target lives in the span of the terminal targets $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$. In Lemma 5.6, we then convert $\boldsymbol{u}_C$ into the fresh outgoing coordinate $\boldsymbol{h}_C$ by adding one always-available connector vector.

Lemma 5.5 (Internal chain invariant)**.** *Let C be a heavy chain with terminal targets $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$, where each $\boldsymbol{z}_i$ is either a leaf target or the helper target of a light child chain. Suppose $\boldsymbol{z}_i$ has balanced witness bound S_i. Then applying the gate-composition construction along C gives a vector-form span program for the formula represented by the chain, with an internal unit target*

$$\boldsymbol{u}_C \in \operatorname{span}\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k\}.$$

The target $\boldsymbol{u}_C$ has witness parameters (A_C, B_C) satisfying

$$(A_C + 1)B_C = \sum_{i=1}^{k} (S_i^2 + S_i).$$

Moreover, $\boldsymbol{u}_C$ is a nonnegative linear combination of $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$. The OR gates along the chain may also introduce always-available rotation vectors: these are the extra vectors $\boldsymbol{r}$ from the OR case of Lemma 5.3 and will be used in the matrix-norm estimate.

Proof. For a terminal target $\boldsymbol{z}_i$, use witness parameters (S_i, S_i). Then the contribution of this terminal to the invariant $(A+1)B$ is

$$(S_i + 1)S_i = S_i^2 + S_i.$$

Set $\boldsymbol{u}_k := \boldsymbol{z}_k$. We now move upward along the heavy chain. For $j = k-1, k-2, \ldots, 1$, assume that $\boldsymbol{u}_{j+1}$ is the target already constructed for the subformula rooted at the heavy child of v_j. The other child of v_j contributes the terminal target $\boldsymbol{z}_j$. Apply Lemma 5.3 to the two child targets $\boldsymbol{u}_{j+1}$ and $\boldsymbol{z}_j$, using the actual gate label at v_j. This produces a new target $\boldsymbol{u}_j$ that computes the subformula rooted at v_j, interpreting each terminal target $\boldsymbol{z}_i$ as the value of its corresponding leaf or light-child subformula.

Repeating this until $j = 1$ gives $\boldsymbol{u}_C := \boldsymbol{u}_1$. At every step, Lemma 5.3 says that the quantity $(A+1)B$ for the newly constructed target is the sum of the corresponding quantities for the two child targets. Hence this quantity accumulates additively over the k terminal targets, and

$$(A_C + 1)B_C = \sum_{i=1}^{k} (S_i^2 + S_i).$$

Since every gate target $\boldsymbol{u}_j$ is chosen as a nonnegative linear combination of its children, $\boldsymbol{u}_C$ is a nonnegative linear combination of the vectors $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$. □

Lemma 5.6 (Helper recurrence)**.** *Let C be a heavy chain with terminal targets $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$, and suppose these terminal targets have balanced witness bounds $S_1, \ldots, S_k$. There is a vector-form span program for the subtree rooted at the top node of C, with unit target $\boldsymbol{h}_C$, all available vectors unit, and balanced witness bound*

$$S_C = 1 + \sqrt{\sum_{i=1}^{k} (S_i^2 + S_i)}.$$

Proof. Let $\boldsymbol{u}_C$ be the internal target from Lemma 5.5, with witness parameters (A_C, B_C). Introduce the fresh unit coordinate $\boldsymbol{h}_C$. Choose $\delta_C \in (0, \pi/2)$, to be specified below, and add the always-available *connector vector*

$$\boldsymbol{g}_C := \cos\delta_C \, \boldsymbol{h}_C - \sin\delta_C \, \boldsymbol{u}_C.$$

This vector is unit, and it is an available vector, not a new coordinate. The target of the chain program is $\boldsymbol{h}_C$. Because $\boldsymbol{g}_C$ is always available and both $\sin\delta_C$ and $\cos\delta_C$ are nonzero, $\boldsymbol{h}_C$ is in the available span if and only if $\boldsymbol{u}_C$ is in the available span. Thus replacing $\boldsymbol{u}_C$ by the helper target $\boldsymbol{h}_C$ preserves the Boolean function computed by the chain.

If the chain evaluates to 1, let $\boldsymbol{a}_C$ be a 1-witness for $\boldsymbol{u}_C$, with $\|\boldsymbol{a}_C\|^2 \le A_C$. Since

$$\boldsymbol{h}_C = \frac{\sin\delta_C}{\cos\delta_C}\boldsymbol{u}_C + \frac{1}{\cos\delta_C}\boldsymbol{g}_C,$$

a 1-witness for $\boldsymbol{h}_C$ is obtained by using $(\sin\delta_C/\cos\delta_C)\boldsymbol{a}_C$ for the internal witness and coefficient $1/\cos\delta_C$ on $\boldsymbol{g}_C$. Its size is at most

$$\frac{A_C \sin^2\delta_C}{\cos^2\delta_C} + \frac{1}{\cos^2\delta_C} = 1 + \frac{(A_C+1)\sin^2\delta_C}{\cos^2\delta_C}.$$

If the chain evaluates to 0, let $\boldsymbol{w}_C$ be a 0-witness for $\boldsymbol{u}_C$, with

$$\boldsymbol{w}_C^\top \boldsymbol{u}_C = 1, \qquad \|\boldsymbol{w}_C\|^2 \le B_C.$$

Regard $\boldsymbol{w}_C$ as a vector in the old ambient space and extend it by zero on the new coordinate $\boldsymbol{h}_C$. Define

$$\boldsymbol{w}'_C := \boldsymbol{h}_C + \frac{\cos\delta_C}{\sin\delta_C}\boldsymbol{w}_C.$$

Then $(\boldsymbol{w}'_C)^\top \boldsymbol{h}_C = 1$, and

$$(\boldsymbol{w}'_C)^\top(\cos\delta_C \boldsymbol{h}_C - \sin\delta_C \boldsymbol{u}_C) = 0.$$

Thus $\boldsymbol{w}'_C$ is orthogonal to the connector vector $\boldsymbol{g}_C$, and it is orthogonal to all available internal vectors because $\boldsymbol{w}_C$ is. Its size is at most

$$1 + \frac{B_C\cos^2\delta_C}{\sin^2\delta_C}.$$

Choose δ_C so that

$$\frac{\sin^2\delta_C}{\cos^2\delta_C} = \sqrt{\frac{B_C}{A_C+1}}.$$

With this choice, the positive and negative witness-size bounds are both equal to

$$1 + \sqrt{(A_C+1)B_C}.$$

Therefore the outgoing target $\boldsymbol{h}_C$ has balanced witness bound

$$S_C = 1 + \sqrt{(A_C+1)B_C}.$$

Using Lemma 5.5, we obtain

$$S_C = 1 + \sqrt{\sum_{i=1}^{k}(S_i^2 + S_i)}.$$

□

5.3 Witness Size at the Root

The recurrence from Lemma 5.6 is local to one heavy chain. We next show that the heavy-light structure turns this local recurrence into a global square-root bound at the root.

Lemma 5.7 (Root witness size)**.** *For every heavy chain C, let n_C be the number of formula leaves below the top node of C. The balanced witness bound S_C produced by Lemma 5.6 satisfies*

$$S_C \le 100\sqrt{n_C} - 5.$$

Consequently, the root target has balanced witness bound $O(\sqrt{n})$.

Proof. The heavy chains themselves form a tree: a chain D is a child of a chain C when the top node of D is a light child of some node on C. We induct over this tree from the leaves upward.

The base case is a chain whose terminal targets are all leaves; the same calculation below applies to this case, using $S_i = 1$ and $n_i = 1$ for every terminal target.

Fix a chain C, with terminal targets $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$. Associate to each terminal target a leaf count n_i: if $\boldsymbol{z}_i$ is a leaf coordinate, set $n_i = 1$; if $\boldsymbol{z}_i = \boldsymbol{h}_D$ for a child chain D, set $n_i = n_D$. The terminal subtrees partition the leaves below the top node of C, so

$$\sum_{i=1}^{k} n_i = n_C.$$

We also have

$$n_i \le \frac{n_C}{2} \qquad \text{for every } i.$$

If $\boldsymbol{z}_i$ is a helper coordinate for a light child, then that light child has at most half the leaves of the internal node where it branches off, and that internal node's subtree has at most n_C leaves. Hence $n_i \le n_C/2$. If $\boldsymbol{z}_i$ is a leaf terminal contributed by a light child, or is the bottom leaf of the heavy path, then $n_i = 1 \le n_C/2$, because C is rooted at an internal node and hence $n_C \ge 2$.

For a leaf terminal target, $S_i = 1 \le 100\sqrt{n_i} - 5$. For a child-chain terminal target, the same inequality holds by induction. Therefore, for every i,

$$S_i \le 100\sqrt{n_i} - 5.$$

Since $n_i \ge 1$, we have

$$k \le \sum_{i=1}^{k} \sqrt{n_i}.$$

Also,

$$\sum_{i=1}^{k} \sqrt{n_i} \ge \frac{\sum_{i=1}^{k} n_i}{\sqrt{\max_i n_i}} \ge \frac{n_C}{\sqrt{n_C/2}} = \sqrt{2n_C}.$$

By Lemma 5.6,

$$S_C = 1 + \sqrt{\sum_{i=1}^{k} (S_i^2 + S_i)}.$$

Using $S_i \le 100\sqrt{n_i} - 5$ and the monotonicity of $t^2 + t$ for $t \ge 0$,

$$\begin{aligned}
\sum_{i=1}^{k} (S_i^2 + S_i) &\le \sum_{i=1}^{k} \left((100\sqrt{n_i} - 5)^2 + (100\sqrt{n_i} - 5)\right) \\
&= 10000n_C - 900 \sum_{i=1}^{k} \sqrt{n_i} + 20k \\
&\le 10000n_C - 880 \sum_{i=1}^{k} \sqrt{n_i} \\
&\le 10000n_C - 880\sqrt{2n_C}.
\end{aligned}$$

As $880\sqrt{2} > 1200$, this is at most

$$10000n_C - 1200\sqrt{n_C} + 36 = (100\sqrt{n_C} - 6)^2.$$

Therefore

$$S_C \le 1 + (100\sqrt{n_C} - 6) = 100\sqrt{n_C} - 5.$$

This closes the induction. For the root chain R, $n_R = n$, so $S_R = O(\sqrt{n})$. $\square$

5.4 The Entrywise-Absolute Matrix Norm

The previous subsection showed that the balanced witness bound remains $O(\sqrt{n})$, even after adding helper coordinates and connector vectors. We now prove the second hypothesis needed by Lemma 4.1: the available-vector matrix has bounded entrywise-absolute operator norm.

From this point on, the available-vector matrix is viewed with available vectors as columns, as in Definition 3.1. Rotation vectors are exactly the additional always-available vectors introduced by OR gates in Lemma 5.3; connector vectors are the always-available vectors $\boldsymbol{g}_C$ introduced in Lemma 5.6.

In Lemma 5.8, we first prove a bound for each individual heavy-chain block. After that we combine the blocks using the fact that every coordinate appears in at most two chain blocks.

Lemma 5.8 (Chain block norm)**.** *For each heavy chain C, let $\boldsymbol{V}_C$ be the matrix whose columns are the available vectors assigned to C, namely the OR-rotation vectors of C and the connector vector $\boldsymbol{g}_C$. Then*

$$\|\mathrm{abs}(\boldsymbol{V}_C)\| = O(1).$$

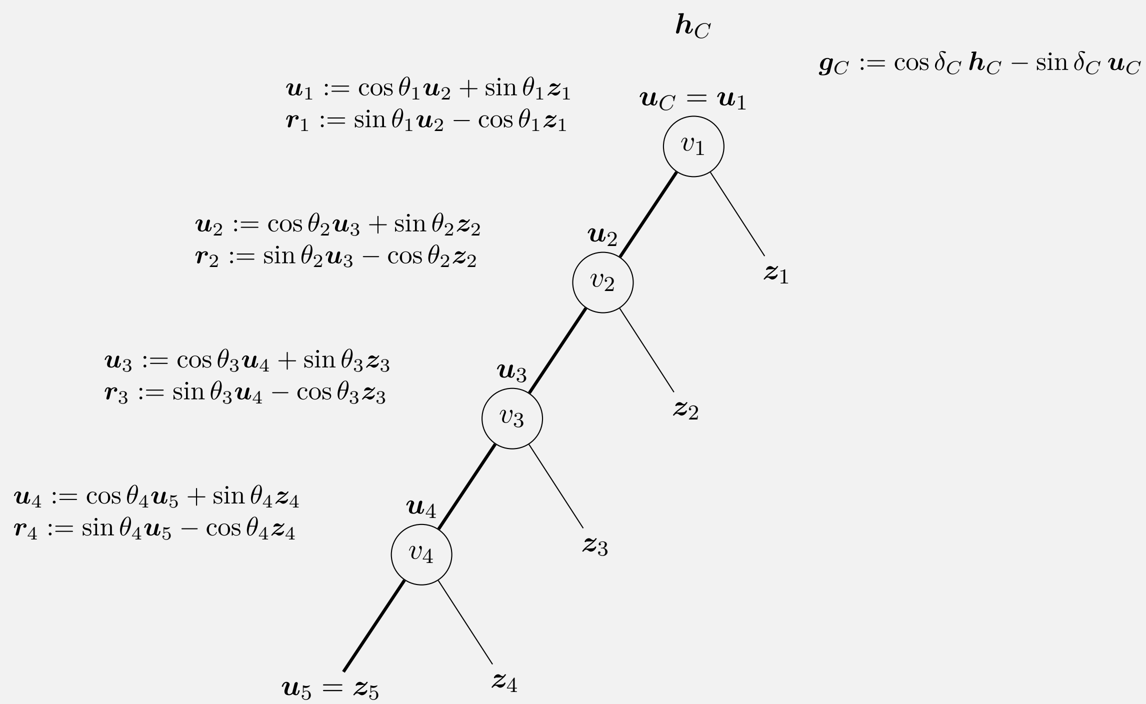

Figure 3: The chain block used in the norm bound. The recursive targets $\boldsymbol{u}_j$ lie on the heavy path, the side targets are $\boldsymbol{z}_j$, and the formal rotation vectors $\boldsymbol{r}_j$ are recorded for the matrix-norm estimate.

Proof. Work in the orthonormal terminal basis $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ for the chain C, ordered from top to bottom along the heavy path. At each node we order the two child targets so that $\boldsymbol{u}_{j+1}$ is the target continuing along the heavy path and $\boldsymbol{z}_j$ is the side terminal. If the actual left-right ordering of an OR gate gives the negative of the displayed rotation vector below, this is harmless: the span and the entrywise absolute values are unchanged. Set $\boldsymbol{u}_k := \boldsymbol{z}_k$. For $j = k-1, k-2, \dots, 1$, the internal target has the form

$$\boldsymbol{u}_j = \cos\theta_j\, \boldsymbol{u}_{j+1} + \sin\theta_j\, \boldsymbol{z}_j, \qquad \theta_j \in [0, \pi/2].$$

Here $\boldsymbol{u}_{j+1}$ is supported on later terminals $\boldsymbol{z}_{j+1}, \dots, \boldsymbol{z}_k$, and hence is orthogonal to $\boldsymbol{z}_j$. The corresponding formal rotation vector is

$$\boldsymbol{r}_j = \sin\theta_j\, \boldsymbol{u}_{j+1} - \cos\theta_j\, \boldsymbol{z}_j.$$

The word "formal" means that for AND gates this vector is not actually inserted into the span program. We include it only for the norm bound; the actual OR-rotation vectors form a submatrix of this formal matrix and therefore have no larger operator norm.

With coordinates split as

$$\mathbb{R}\boldsymbol{z}_j \oplus^{\perp} \operatorname{span}\{\boldsymbol{z}_{j+1}, \dots, \boldsymbol{z}_k\},$$

the two vectors may be viewed as

$$\boldsymbol{u}_j = \begin{pmatrix} \sin\theta_j \\ \cos\theta_j\, \boldsymbol{u}_{j+1} \end{pmatrix}, \qquad \boldsymbol{r}_j = \begin{pmatrix} -\cos\theta_j \\ \sin\theta_j\, \boldsymbol{u}_{j+1} \end{pmatrix}.$$

We claim that the formal rotation vectors $\boldsymbol{r}_1, \dots, \boldsymbol{r}_{k-1}$ are orthonormal. We prove this by downward induction on j. The induction statement is that

$$\boldsymbol{u}_j, \boldsymbol{r}_j, \boldsymbol{r}_{j+1}, \dots, \boldsymbol{r}_{k-1}$$

form an orthonormal basis of $\operatorname{span}\{\boldsymbol{z}_j, \boldsymbol{z}_{j+1}, \dots, \boldsymbol{z}_k\}$, with the convention that the list of rotation vectors is empty when $j = k$. The base case is $j = k$, where $\boldsymbol{u}_k = \boldsymbol{z}_k$.

For the induction step, assume that the list

$$\boldsymbol{u}_{j+1}, \boldsymbol{r}_{j+1}, \dots, \boldsymbol{r}_{k-1}$$

is an orthonormal basis of $\operatorname{span}\{\boldsymbol{z}_{j+1}, \dots, \boldsymbol{z}_k\}$. Since $\boldsymbol{z}_j$ is a unit vector orthogonal to that span, the vectors

$$\boldsymbol{z}_j, \boldsymbol{u}_{j+1}, \boldsymbol{r}_{j+1}, \dots, \boldsymbol{r}_{k-1}$$

are orthonormal and span $\operatorname{span}\{\boldsymbol{z}_j, \dots, \boldsymbol{z}_k\}$. The pair $\boldsymbol{u}_j, \boldsymbol{r}_j$ is obtained from the pair $\boldsymbol{u}_{j+1}, \boldsymbol{z}_j$ by the orthogonal two-dimensional change of basis

$$\begin{pmatrix} \boldsymbol{u}_j \\ \boldsymbol{r}_j \end{pmatrix} = \begin{pmatrix} \cos\theta_j & \sin\theta_j \\ \sin\theta_j & -\cos\theta_j \end{pmatrix} \begin{pmatrix} \boldsymbol{u}_{j+1} \\ \boldsymbol{z}_j \end{pmatrix}.$$

Thus replacing $\boldsymbol{u}_{j+1}, \boldsymbol{z}_j$ by $\boldsymbol{u}_j, \boldsymbol{r}_j$ preserves orthonormality and gives an orthonormal basis of $\operatorname{span}\{\boldsymbol{z}_j, \dots, \boldsymbol{z}_k\}$. This proves the claim. Let $\boldsymbol{R}$ be the matrix with columns $\boldsymbol{r}_j$. Then $\|\boldsymbol{R}\| \le 1$.

Because every internal target $\boldsymbol{u}_j$ is built using nonnegative coefficients, $\boldsymbol{u}_{j+1}$ has nonnegative coordinates in the terminal basis and is supported only on $\boldsymbol{z}_{j+1}, \dots, \boldsymbol{z}_k$. The vector $\boldsymbol{z}_j$ is a distinct terminal coordinate. Therefore taking entrywise absolute values only changes the sign of the $\boldsymbol{z}_j$ term:

$$\operatorname{abs}(\boldsymbol{r}_j) = \sin\theta_j\, \boldsymbol{u}_{j+1} + \cos\theta_j\, \boldsymbol{z}_j = \boldsymbol{r}_j + 2\cos\theta_j\, \boldsymbol{z}_j.$$

Thus

$$\operatorname{abs}(\boldsymbol{R}) = \boldsymbol{R} + \boldsymbol{D},$$

where the j-th column of $\boldsymbol{D}$ is $2\cos\theta_j\, \boldsymbol{z}_j$. In the terminal basis, this matrix has the schematic form

$$\boldsymbol{D} = \begin{pmatrix} 2\cos\theta_1 & 0 & \cdots & 0 \\ 0 & 2\cos\theta_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 2\cos\theta_{k-1} \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Since $2\cos\theta_j \le 2$, we have $\|\boldsymbol{D}\| \le 2$. Therefore

$$\|\mathrm{abs}(\boldsymbol{R})\| \le \|\boldsymbol{R}\| + \|\boldsymbol{D}\| \le 3.$$

The connector vector satisfies

$$\boldsymbol{g}_C = \cos\delta_C\, \boldsymbol{h}_C - \sin\delta_C\, \boldsymbol{u}_C.$$

Since $\boldsymbol{h}_C \perp \boldsymbol{u}_C$ and both are unit,

$$\mathrm{abs}(\boldsymbol{g}_C) = \cos\delta_C\, \boldsymbol{h}_C + \sin\delta_C\, \boldsymbol{u}_C$$

is also unit. Therefore the chain block, consisting of the actual OR-rotation columns and the connector column, satisfies

$$\left\| \begin{pmatrix} \mathrm{abs}(\boldsymbol{R}) & \mathrm{abs}(\boldsymbol{g}_C) \end{pmatrix} \right\| \le \|\mathrm{abs}(\boldsymbol{R})\| + \|\mathrm{abs}(\boldsymbol{g}_C)\| \le 4.$$

In particular, $\|\mathrm{abs}(\boldsymbol{V}_C)\| = O(1)$. □

The local bound from Lemma 5.8 would not be enough if a coordinate appeared in many chain blocks. The helper construction avoids this: every coordinate is used locally in its own chain and at most once by its parent chain, so each coordinate appears in at most two chain blocks.

Lemma 5.9 (Global absolute norm)**.** *Let $\boldsymbol{V}$ be the full available-vector matrix of the span program, with available vectors as columns. Then*

$$\|\mathrm{abs}(\boldsymbol{V})\| = O(1).$$

Proof. Let $\boldsymbol{V}_{\mathrm{lit}}$ be the submatrix of literal columns. These columns are distinct leaf-coordinate basis vectors, so $\|\mathrm{abs}(\boldsymbol{V}_{\mathrm{lit}})\| \le 1$.

Assign each available vector introduced by a chain to the unique chain in which it is introduced: OR-rotation vectors are assigned to their chain, and the connector vector $\boldsymbol{g}_C$ is assigned to C. Let $\boldsymbol{V}_C$ be the resulting block of columns for chain C. By Lemma 5.8, there is an absolute constant c_{block} such that

$$\|\mathrm{abs}(\boldsymbol{V}_C)\boldsymbol{x}\| \le c_{\mathrm{block}}\, \|\boldsymbol{x}\|$$

for every coefficient vector $\boldsymbol{x}$ on the columns of $\boldsymbol{V}_C$.

Let $\mathrm{coords}(C)$ be the set of ambient coordinates (i.e., rows) that appear in the columns of $\boldsymbol{V}_C$. A helper coordinate $\boldsymbol{h}_C$ appears in the block of C, and possibly also in the parent block where it is used as a terminal target. A leaf coordinate can appear in the block of only the chain where that leaf is a terminal target. Thus every coordinate belongs to at most two such sets. Consequently, for any ambient vector $\boldsymbol{y}$,

$$\sum_C \left\| \boldsymbol{y}|_{\mathrm{coords}(C)} \right\|^2 \le 2\, \|\boldsymbol{y}\|^2.$$

For a global coefficient vector $\boldsymbol{x}$, write $\boldsymbol{x}_{\text{lit}}$ for its restriction to literal columns and $\boldsymbol{x}_C$ for its restriction to the columns assigned to C. Set

$$\boldsymbol{y}_{\text{lit}} := \text{abs}(\boldsymbol{V}_{\text{lit}})\boldsymbol{x}_{\text{lit}}, \qquad \boldsymbol{y}_C := \text{abs}(\boldsymbol{V}_C)\boldsymbol{x}_C.$$

The vectors $\boldsymbol{y}_C$ are supported only on coords(C). Since every ambient coordinate belongs to at most two sets coords(C),

$$\left\| \sum_C \boldsymbol{y}_C \right\|^2 \leq 2 \sum_C \|\boldsymbol{y}_C\|^2 .$$

Also $\|\boldsymbol{y}_{\text{lit}}\| \leq \|\boldsymbol{x}_{\text{lit}}\|$, because $\boldsymbol{V}_{\text{lit}}$ is a submatrix of the identity. The vectors $\boldsymbol{x}_{\text{lit}}$ and $\boldsymbol{x}_C$ live in disjoint coefficient-coordinate sets, namely the literal columns and the columns assigned to the chain blocks, so their squared norms add inside $\|\boldsymbol{x}\|^2$. Their images $\boldsymbol{y}_{\text{lit}}$ and $\boldsymbol{y}_C$ live in the ambient coordinate space and need not be mutually orthogonal: $\boldsymbol{y}_{\text{lit}}$ is supported on leaf coordinates, while $\boldsymbol{y}_C$ is supported on coords(C). A leaf coordinate may therefore appear in $\boldsymbol{y}_{\text{lit}}$ and in one chain image, and a helper coordinate may appear in two chain images. This is why the estimate below uses the crude inequality $\|\boldsymbol{a}+\boldsymbol{b}\|^2 \leq 2\|\boldsymbol{a}\|^2 + 2\|\boldsymbol{b}\|^2$, together with the bounded-overlap estimate for the $\boldsymbol{y}_C$'s. Therefore

$$\begin{aligned}
\|\text{abs}(\boldsymbol{V})\boldsymbol{x}\|^2 &= \left\| \boldsymbol{y}_{\text{lit}} + \sum_C \boldsymbol{y}_C \right\|^2 \\
&\leq 2\|\boldsymbol{y}_{\text{lit}}\|^2 + 2\left\| \sum_C \boldsymbol{y}_C \right\|^2 \\
&\leq 2\|\boldsymbol{x}_{\text{lit}}\|^2 + 4\sum_C \|\boldsymbol{y}_C\|^2 \\
&\leq 2\|\boldsymbol{x}_{\text{lit}}\|^2 + 4c_{\text{block}}^2 \sum_C \|\boldsymbol{x}_C\|^2 \\
&\leq (2 + 4c_{\text{block}}^2)\|\boldsymbol{x}\|^2 .
\end{aligned}$$

Therefore $\|\text{abs}(\boldsymbol{V})\| = O(1)$. □

5.5 Completion of the Proof

Proof of Theorem 5.1. If $n = 1$, the theorem follows directly from the leaf polynomial $P(x) = x_i$ or $P(x) = 1 - x_i$, according to the literal at the single leaf. Hence assume $n \geq 2$. Let R be the root heavy chain. By Lemma 5.7, the root helper target $\boldsymbol{h}_R$ has balanced witness bound

$$S_R = O(\sqrt{n}).$$

The root target $\boldsymbol{h}_R$ computes the formal occurrence-variable function $F(y)$: leaf programs compute leaves, Lemma 5.3 preserves the gate semantics, and each helper connector preserves the available span equivalence between $\boldsymbol{u}_C$ and $\boldsymbol{h}_C$. By construction, $\boldsymbol{h}_R$ is a unit vector and every available vector is unit. By Lemma 5.9, the available-vector matrix $\boldsymbol{V}$ satisfies

$$\|\text{abs}(\boldsymbol{V})\| = O(1).$$

Lemma 4.1 gives a polynomial P in the formal occurrence variables satisfying

$$|P(y) - F(y)| \leq \frac{1}{3}$$

for every Boolean input y, with

$$\deg P = O(S_R) = O(\sqrt{n}), \qquad \|P\|_1 \le 2^{O(S_R)} = 2^{O(\sqrt{n})}.$$

For the original formula $F : \{0,1\}^m \to \{0,1\}$, substitute each formal occurrence variable y_ℓ by the literal x_i or $1 - x_i$ labeling that leaf. A degree-d monomial expands with coefficient ℓ_1-norm at most 2^d, and here $d = O(\sqrt{n})$, so this loss is absorbed into $2^{O(\sqrt{n})}$. Identifying repeated positive variables and multilinearizing on the Boolean cube only merge coefficients of identical monomials, so they cannot increase the coefficient ℓ_1-norm. The resulting polynomial is in the original m variables and satisfies the theorem.

This proves Theorem 5.1. □

6 Tightness of the ℓ_1-Norm Bound

In this section we show that the coefficient bound in Theorem 5.1 is tight up to the constant in the exponent. The example is the De Morgan formula for inner product modulo 2.

Theorem 6.1 (Tightness of the coefficient bound)**.** *For all sufficiently large n, there is a Boolean function F computable by a De Morgan formula with at most n leaves such that every real polynomial P satisfying $|P(z) - F(z)| \le 1/3$ for every Boolean input z also satisfies $\|P\|_1 \ge 2^{\Omega(\sqrt{n})}$. This lower bound holds with no restriction on the degree of P.*

Proof. Let

$$\mathrm{IP}_t(x,y) := \bigoplus_{i=1}^{t} x_i y_i, \qquad x, y \in \{0,1\}^t.$$

There is an absolute constant C such that IP_t has a De Morgan formula with at most Ct^2 leaves: parity on t inputs has an $O(t^2)$-leaf De Morgan formula, obtained by the standard balanced recurrence for parity and its negation, and each parity input can be replaced by $x_i \wedge y_i$ or its negation.

Let

$$f_t(x,y) := (-1)^{\mathrm{IP}_t(x,y)}$$

be the sign version. With respect to the normalized Fourier basis $\chi_{a,b}(x,y) := (-1)^{a\cdot x + b\cdot y}$, where $a, b \in \{0,1\}^t$, we have

$$\widehat{f_t}(a,b) = \mathbf{E}_{x,y}\Big[(-1)^{x\cdot y + a\cdot x + b\cdot y}\Big] = 2^{-t}(-1)^{a\cdot b}.$$

Thus every Fourier coefficient of f_t has magnitude 2^{-t}.

Now let P be any real polynomial satisfying $|P(z) - \mathrm{IP}_t(z)| \le 1/3$ on the Boolean cube, and set $Q := 1 - 2P$. Since $f_t = 1$ on $\mathrm{IP}_t = 0$ inputs and $f_t = -1$ on $\mathrm{IP}_t = 1$ inputs, the approximation guarantee gives

$$f_t(z)Q(z) \ge \frac{1}{3} \qquad \text{for every Boolean input } z.$$

Therefore

$$\frac{1}{3} \le \mathbf{E}[f_t Q] = \sum_S \widehat{f_t}(S)\widehat{Q}(S) \le 2^{-t} \sum_S |\widehat{Q}(S)|.$$

So the Fourier coefficient ℓ_1-norm of Q is at least $2^t/3$.

It remains to transfer this lower bound to the $\{0,1\}$-monomial basis used in the theorem. For a monomial $M_T(z) = \prod_{i \in T} z_i$,

$$M_T(z) = 2^{-|T|} \sum_{R \subseteq T} (-1)^{|R|} \chi_R(z),$$

so every $\{0,1\}$-monomial, equivalently every AND of variables, has Fourier ℓ_1-norm exactly 1. Hence $\left\|\widehat{H}\right\|_1 \leq \|H\|_1$ for every polynomial H written in the $\{0,1\}$-monomial basis. Applying this to $Q = 1 - 2P$,

$$\frac{2^t}{3} \leq \left\|\widehat{Q}\right\|_1 \leq \|Q\|_1 \leq 1 + 2\,\|P\|_1\,.$$

Thus $\|P\|_1 \geq 2^{\Omega(t)}$. Finally, choose $t = \lfloor \sqrt{n/C} \rfloor$ and let $F = \mathrm{IP}_t$. Then F has a De Morgan formula with at most n leaves and every pointwise approximant satisfies $\|P\|_1 \geq 2^{\Omega(\sqrt{n})}$. $\square$

Acknowledgments

The author is thankful to Avishay Tal for valuable discussions.

A Appendix – Statement on the Use of AI

We used GPT-5.5 Pro during the development and writing of this work. Its most substantial contribution concerns the proof in Section 5. We describe the division of work below.

The proof begins by considering two special cases:

- The special case in which the formula is a single long chain was originally solved by the human author. Its span-program construction is the repeated gate-composition construction formalized in Lemmas 5.3 and 5.5. GPT-5.5 Pro later simplified the proof of the entrywise-absolute norm bound for this case; the resulting argument appears in Lemma 5.8. The underlying span-program construction remained unchanged.

- The special case of a balanced alternating AND–OR tree was obtained by GPT-5.5 Pro. Its argument is equivalent to placing a helper vector at every internal node. The proof cleverly combines ideas from the formula-evaluation construction of Ambainis et al. [ACR+10] and the span-program framework developed by Reichardt [Rei09, Rei11, Rei12], but the resulting construction and analysis are not the same as either previous argument.

The extension from these two special cases to general formulas, via the heavy-chain decomposition used in Section 5, was carried out by the human author. This step leads to the complete construction and analysis in Lemmas 5.6, 5.7 and 5.9. GPT-5.5 Pro did not succeed in completing this step, even when explicitly prompted to use a heavy-chain decomposition. Its attempts typically became confused about the normalization of available vectors, the scaling of witnesses associated with different subtrees, or the interaction between these choices. More generally, the model tended to pursue local inductive arguments without maintaining a sufficiently global perspective on the overall construction. In contrast, for human, this part was mostly a matter of careful engineering rather than requiring particularly deep new ideas, and it was completed over approximately one week of work.

GPT-5.5 Pro was also used to automate and accelerate certain self-contained substeps of roughly IMO difficulty. For example, it wrote up some of the recurrence calculations in the witness-size analysis in Lemma 5.7. It was also used to simplify proofs and check calculations throughout the manuscript.

All AI-generated mathematical content was reviewed and, when necessary, corrected or rewritten by the human author. The human author made the final decisions about the statements, constructions, proofs, and presentation, and takes responsibility for the correctness of the paper.

References

[ACR+10] Andris Ambainis, Andrew M. Childs, Ben W. Reichardt, Robert Špalek, and Shengyu 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.

[BBC+01] 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.

[BDBGK18] Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 339–349. IEEE Computer Society, 2018.

[BKT21] Mark Bun, Robin Kothari, and Justin Thaler. Quantum algorithms and approximating polynomials for composed functions with shared inputs. *Quantum*, 5:543, 2021.

[BNRdW07] Harry Buhrman, Ilan Newman, Hein R"ohrig, and Ronald de Wolf. Robust polynomials and quantum algorithms. *Theory of Computing Systems*, 40(4):379–395, 2007.

[Bog18] Andrej Bogdanov. Small bias requires large formulas. In *45th International Colloquium on Automata, Languages, and Programming, ICALP 2018*, volume 107 of *LIPIcs*, pages 22:1–22:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018.

[BT20] Mark Bun and Justin Thaler. Approximate degree in classical and quantum computing. *SIGACT News*, 51(4):48–72, 2020.

[BW24] Gabriel Bathie and R. Ryan Williams. Towards stronger depth lower bounds. In *15th Innovations in Theoretical Computer Science Conference, ITCS 2024*, volume 287 of *LIPIcs*, pages 10:1–10:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

[CDL26] Arkadev Chattopadhyay, Yogesh Dahiya, and Shachar Lovett. Restriction trees for sparsity and applications. In *Proceedings of the 58th Annual ACM Symposium on Theory of Computing*, pages 1892–1900. ACM, 2026.

[Che18] Lijie Chen. Personal communication on recursive robust polynomial approximations for formulas, 2018. Personal communication.

[DM16] Irit Dinur and Or Meir. Toward the KRW composition conjecture: Cubic formula lower bounds via communication complexity. In *31st Conference on Computational Complexity, CCC 2016*, volume 50 of *LIPIcs*, pages 3:1–3:51. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016.

[FMT23] Yuval Filmus, Or Meir, and Avishay Tal. Shrinkage under random projections, and cubic formula lower bounds for AC0. *Theory of Computing*, 19:1–51, 2023.

[GMWW14] Dmitry Gavinsky, Or Meir, Omri Weinstein, and Avi Wigderson. Toward better formula lower bounds: An information complexity approach to the KRW composition conjecture. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing, STOC 2014*, pages 213–222. ACM, 2014.

[Hås98] Johan Håstad. The shrinkage exponent of De Morgan formulae is 2. *SIAM Journal on Computing*, 27(1):48–64, 1998.

[HHTT21] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Fooling constant-depth threshold circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021*, pages 104–115. IEEE, 2021.

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012*, pages 111–119. IEEE Computer Society, 2012.

[Khr71] V. M. Khrapchenko. Complexity of the realization of a linear function in the case of pi-circuits. *Mathematical Notes of the Academy of Sciences of the USSR*, 9:21–23, 1971.

[KKL+21] Valentine Kabanets, Sajin Koroth, Zhenjian Lu, Dimitrios Myrisiotis, and Igor C. Oliveira. Algorithms and lower bounds for De Morgan formulas of low-communication leaf gates. *ACM Transactions on Computation Theory*, 13(4):23:1–23:37, 2021.

[KR13] Ilan Komargodski and Ran Raz. Average-case lower bounds for formula size. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing, STOC 2013*, pages 171–180. ACM, 2013.

[KRT17] Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for De Morgan formula size: Matching worst-case lower bound. *SIAM Journal on Computing*, 46(1):37–57, 2017.

[KRW95] Mauricio Karchmer, Ran Raz, and Avi Wigderson. Super-logarithmic depth lower bounds via the direct sum in communication complexity. *Computational Complexity*, 5(3/4):191–204, 1995.

[KW88] Mauricio Karchmer and Avi Wigderson. Monotone circuits for connectivity require super-logarithmic depth. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, STOC 1988*, pages 539–550. ACM, 1988.

[LLS06] Sophie Laplante, Troy Lee, and Mario Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15(2):163–196, 2006.

[Mei23] Or Meir. Toward better depth lower bounds: A KRW-like theorem for strong composition. In *64th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2023*, pages 1056–1081. IEEE Computer Society, 2023.

[MS21] Ivan Mihajlin and Alexander Smal. Toward better depth lower bounds: The XOR-KRW conjecture. In *36th Computational Complexity Conference, CCC 2021*, volume 200 of *LIPIcs*, pages 38:1–38:24. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021.

[NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4:301–313, 1994.

[Rei09] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009*, pages 544–551. IEEE Computer Society, 2009.

[Rei11] Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011*, pages 560–569. SIAM, 2011.

[Rei12] Ben W. Reichardt. Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8(13):291–319, 2012.

[She08] Alexander A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, STOC 2008*, pages 85–94. ACM, 2008.

[She13] Alexander A. Sherstov. Making polynomials robust to noise. *Theory of Computing*, 9:593–615, 2013.

[Tal14] Avishay Tal. Shrinkage of De Morgan formulae by spectral techniques. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014*, pages 551–560. IEEE Computer Society, 2014.

[Tal17] Avishay Tal. Formula lower bounds via the quantum method. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017*, pages 1256–1268. ACM, 2017.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il