THEORY OF COMPUTING, Volume 14 (12), 2018, pp. 1–24
www.theoryofcomputing.org

Certifying Polynomials for $\mathsf{AC}^0[\oplus]$ Circuits, with Applications to Lower Bounds and Circuit Compression

Swastik Kopparty* Srikanth Srinivasan†

Received April 18, 2016; Revised March 20, 2017; Published August 25, 2018

Abstract: In this paper, we study the method of certifying polynomials for proving $\mathsf{AC}^0[\oplus]$ circuit lower bounds. We use this method to show that Approximate Majority on n bits cannot be computed by $\mathsf{AC}^0[\oplus]$ circuits of size $n^{1+o(1)}$. This implies a separation between the power of $\mathsf{AC}^0[\oplus]$ circuits of near-linear size and uniform $\mathsf{AC}^0[\oplus]$ (and even AC^0) circuits of polynomial size. This also implies a separation between randomized $\mathsf{AC}^0[\oplus]$ circuits of linear size and deterministic $\mathsf{AC}^0[\oplus]$ circuits of near-linear size.

Our proof using certifying polynomials extends the deterministic restrictions technique of Chaudhuri and Radhakrishnan (STOC 1996), who showed that Approximate Majority cannot be computed by AC^0 circuits of size $n^{1+o(1)}$. At the technical level, we show that for every $\mathsf{AC}^0[\oplus]$ circuit C of near-linear size, there is a low-degree polynomial P over $\mathbb{F}_2$ such that the restriction of C to the support of P is constant.

Preliminary versions of this paper appeared in FSTTCS 2012 [19] and on ECCC [29].

*Supported in part by a Sloan Fellowship and NSF grant CCF-1253886.

†Supported by DST Inspire grant IFA12-ENG14. Work partially done when the author was a postdoc at IAS, Princeton and DIMACS, Rutgers University.

ACM Classification: F.1.3

AMS Classification: 68Q17,68Q15

Key words and phrases: complexity theory, circuit complexity, Boolean complexity, polynomial approximation, lower bounds, majority

 DOI: 10.4086/toc.2018.v014a012

We also prove other results exploring various aspects of the power of certifying polynomials. In the process, we show an essentially optimal lower bound of $\log^{\Theta(d)} s \cdot \log(1/\varepsilon)$ on the degree of ε-error approximating polynomials for $\mathsf{AC}^0[\oplus]$ circuits of size s and depth d.

Finally, we use these ideas to give a compression algorithm for $\mathsf{AC}^0[\oplus]$ circuits, answering an open question of Chen, Kabanets, Kolokolova, Shaltiel, and Zuckerman (Computational Complexity 2015).

1 Introduction

In this paper, we study the method of certifying polynomials for proving circuit lower bounds. We begin by describing the motivation for the main new circuit lower bound that we show, after which we will elaborate on the the method itself, and finally we describe some other results exploring the power and limitations of this method.

1.1 The size-hierarchy problem for $\mathsf{AC}^0[\oplus]$

Our main result fits in the general theme of studying the relative power of constant-depth circuit classes. We show a near-tight circuit lower bound for computing Approximate Majority with AND, OR, PARITY and NOT gates. This is a first step in the direction of a uniform size-hierarchy theorem for such circuits, which is a basic open question about this well-studied class of circuits.

We first fix some notation and conventions regarding circuits for the rest of this paper. AC^0 denotes the class of constant-depth circuits with unbounded fan-in AND, OR and NOT gates. $\mathsf{AC}^0[\oplus]$ denotes the class of constant-depth circuits with unbounded fan-in AND, OR, PARITY and NOT gates. We measure the *size* of a circuit by the number of *gates*.[1] We use n to denote the number of input bits to a circuit.

There is a well-developed theory giving superpolynomial and even exponential $(\exp(n^{\Omega(1)}))$ lower bounds for AC^0 and $\mathsf{AC}^0[\oplus]$ circuits [14, 1, 34, 18, 24, 27]. Our focus in this paper is on complexity theory within these classes.

An influential paper of Ragde and Wigderson [23] asked if uniform AC^0 circuits of linear size are strictly weaker than uniform AC^0 circuits of polynomial size. This was answered by Chaudhuri and Radhakrishnan [11], who showed that Approximate Majority functions do not have AC^0 circuits of depth d and near-linear size $O(n^{1+\varepsilon_d})$ (where $\varepsilon_d > 0$). An Approximate Majority function is any function which maps strings of Hamming weight $< n/4$ to 0 and strings of Hamming weight $> 3n/4$ to 1. Such functions were first considered by Ajtai and Ben-Or [2] in the context of AC^0 for the purpose of error-reduction for AC^0 circuits. In [2], it was shown that Approximate Majority can be computed by polynomial-size AC^0 circuits, and later results of Ajtai [1] and Viola [31] showed that this can even be done by uniform polynomial-size AC^0 circuits of depth 3. (In fact these circuits can be made to have depth d and size $O(n^{1+\varepsilon_d})$, where $\varepsilon_d \to 0$ as $d \to \infty$ [11].) This combined with the lower-bound of [11] showed the conjectured separation of Ragde and Wigderson.

The proof method of [11] is especially interesting to us, and we will discuss their method and our extension of it in the next subsection.

[1] It is also standard to use "size" to denote the number of *wires* in the circuit. Our lower bound results also hold for this measure, since the number of gates lower bounds the number of wires up to a constant factor.

A beautiful recent result of Rossman [26] shows a size-hierarchy for AC^0: for every integer $k > 0$, uniform depth-2 AC^0 circuits of size $O(n^k)$ are more powerful than non-uniform AC^0 circuits of size $O(n^{k/4})$ and any constant depth. A striking follow-up result of Amano [4] in fact shows that depth-2 size $O(n^k)$ uniform AC^0 circuits can be more powerful than size $O(n^{k-\varepsilon})$ AC^0 circuits for arbitrary $\varepsilon > 0$ and any constant depth.

In this paper we study the analogous questions for uniform $\mathsf{AC}^0[\oplus]$. Our main result is that Approximate Majority cannot be computed by $\mathsf{AC}^0[\oplus]$ circuits of near-linear size. In particular this means that polynomial-size uniform $\mathsf{AC}^0[\oplus]$ circuits (and even polynomial-size uniform AC^0 circuits) can be more powerful than near-linear size $\mathsf{AC}^0[\oplus]$ circuits. Thus we make a first step towards a size-hierarchy theorem for $\mathsf{AC}^0[\oplus]$ circuits, analogous to the result of Chaudhuri and Radhakrishnan for AC^0. Our result also shows that randomized $\mathsf{AC}^0[\oplus]$ circuits of linear size can be more powerful than deterministic $\mathsf{AC}^0[\oplus]$ circuits of near-linear size.

Showing the full size-hierarchy for uniform $\mathsf{AC}^0[\oplus]$ is still open and would be very interesting. Even the question of whether there exists a function that has uniform $\mathsf{AC}^0[\oplus]$ circuits of size $n^{\log n}$ but no polynomial-size $\mathsf{AC}^0[\oplus]$ circuits (of possibly larger, but constant, depth) remains unanswered.

1.2 Towards a size-hierarchy using certifying polynomials for $\mathsf{AC}^0[\oplus]$

The main component of the [11] lower bound for Approximate Majority is a structure theorem for AC^0 circuits of near-linear size. It states that for every AC^0 circuit C of near-linear size, there is a collection of $o(n)$ variables and and a setting of these variables (a $\{0,1\}$-assignment to them) that simplifies the circuit C to a constant. Equivalently, there is a large axis-parallel subcube of $\{0,1\}^n$ on which C restricts to a constant. This structure theorem immediately implies the lower bound on Approximate Majority.

The proof of this structure theorem is by "deterministic restrictions." Going through the circuit in a bottom-up fashion, one first finds a setting of a small number of variables that simplifies the circuit to one where all the gates have small fan-in. The basic observation is that if one considers the gates at height 1 that have large fan-in, then we can set a large number of them to constants by setting a few input variables; continuing in this way, we eventually remove all large fan-in gates of height 1 (there can't be too many of them, since C is of near-linear size), setting only a few variables in doing so. We then move on to higher levels and repeat the process, which now becomes feasible since setting gates of small fan-in to a constant reduces to setting only a few variables to constants. Once all the gates have small fan-in, the entire circuit is a function of only a few variables and hence, on setting these variables, say, to zero, the circuit becomes a constant.

The main component of our lower bound is an analogous structure theorem for $\mathsf{AC}^0[\oplus]$. Clearly, the structure theorem for AC^0 is false for even a single parity gate and hence for $\mathsf{AC}^0[\oplus]$. However, here we can show that for any $\mathsf{AC}^0[\oplus]$ circuit C of near-linear size, there is a polynomial of degree $o(n)$ such that C restricts to a constant on the support (i. e., non-zero set) of that polynomial. We call such a polynomial a *certifying polynomial* for the circuit C; similar notions have appeared in the complexity theory literature in works of Aspnes, Beigel, Furst and Rudich [7], Green [15], and Alekhnovich and Razborov [3] and also in the context of cryptography in the work of Carlet, Dalai, Gupta, and Maitra [9].

The proof of this structure theorem again proceeds in a bottom-up fashion, but now, instead of setting individual variables to constants, we set low-degree polynomials to constants, thus obtaining a system of polynomial constraints. The polynomials are chosen carefully to ensure that on inputs from the (non-empty) solution set of these constraints, the circuit is equivalent to another circuit all of whose AND and OR gates have small fan-in. Again, once all the AND and OR gates have small fan-in, it is easy to see that the circuit just computes a low-degree polynomial, and thus setting this low-degree polynomial to a constant simplifies the circuit to a constant.

Given this structure theorem, it remains to see that no Approximate Majority function has this structure. This turns out to be a consequence of the general fact that a nonzero polynomial of degree d cannot vanish at every point of a Hamming ball of radius d. (This follows from the fact that Hamming balls are interpolating sets for polynomials.) We therefore conclude that an Approximate Majority function cannot be constant on the zero set of a nonzero polynomial of degree $< n/4$. Combined with the structure theorem, this completes the proof of the lower bound for the $\mathsf{AC}^0[\oplus]$ complexity of Approximate Majority.

1.3 More applications of certifying polynomials

Having proved the lower bound, we then take a step back to re-examine the technique of proving lower bounds via certifying polynomials.

Connection with the Razborov-Smolensky technique. On the face of it, it seems like this method is somewhat distinct from the Razborov-Smolensky method [24, 28] used to prove lower bounds for general $\mathsf{AC}^0[\oplus]$ circuits, which uses *polynomial approximations* to circuits. The Razborov-Smolensky method gives global, approximate structure: it shows that for any $\mathsf{AC}^0[\oplus]$ circuit C of size M, there is a polynomial of degree $\mathrm{poly}(\log(M))$ which agrees with C on most points of $\{0,1\}^n$. Our structure theorem, which only applies to circuits of near-linear size, gives local, exact structure: we get a perfect description of the values taken by an $\mathsf{AC}^0[\oplus]$ circuit on a small but structured subset of $\{0,1\}^n$.

As it turns out, however, the framework of certifying polynomials is quite robust. We demonstrate a connection between polynomial approximations and certifying polynomials for circuits. We then use this connection along with Razborov's approximating polynomials to construct certifying polynomials for general $\mathsf{AC}^0[\oplus]$ circuits. These polynomials have degree much larger than that obtained in our structure theorem, but nevertheless, their degree is small enough to to permit to recover the exponential lower bound obtained by Razborov [24] for $\mathsf{AC}^0[\oplus]$ circuits computing the Majority function. We stress that most of the ideas of this lower-bound proof are already present in [24, 28], and the main aim of this exercise is to show that the use of certifying polynomials is a unified framework that "explains" both our approach and the standard Razborov approach to proving lower bounds for $\mathsf{AC}^0[\oplus]$.

In the course of this proof, we also construct improved approximations to $\mathsf{AC}^0[\oplus]$ circuits in the small-error regime; to the best of our knowledge, such approximations were not known before. Further, since the publication of a preliminary version of this paper [19], this result has been used in other work (see below).

Lower bounds for polynomial approximations. We also exploit the connection between certifying polynomials and polynomial approximations in the reverse direction to prove limits on the power of polynomial approximations. We show that the low-error approximations we construct for $\mathsf{AC}^0[\oplus]$ are close to the best possible for all depths $d \geq 3$. Once again, this demonstrates the usefulness of the certifying polynomials framework.

Compression algorithms for $\mathsf{AC}^0[\oplus]$. A recent paper of Chen et al. [12] introduced the *Compression problem* for a class $\mathcal{C}$ of circuits, which is roughly defined to be the following algorithmic question. Given the truth table of a function f that has a (small) circuit from the class $\mathcal{C}$, the output of the algorithm should be a non-trivially small Boolean circuit (not necessarily from $\mathcal{C}$) computing the function f. The result of Chen et al. [12] shows that this meta-algorithmic question has connections to proving lower bounds for the class $\mathcal{C}$ of circuits. Such algorithms were also shown to exist for AC^0 and some other circuit classes, but the question of whether such an algorithm exists for $\mathsf{AC}^0[\oplus]$ was left open.

Intuitively, a low-degree certifying polynomial for a function f should be useful for this purpose since it computes the function correctly on a certain subset of its inputs, and moreover, it can be computed efficiently by solving a system of linear equations. If we could find "few" certifying polynomials that together "cover" all the inputs of f, then we could use these to give a small circuit for f.

Indeed, we are able to do this by using a recent result due to Nie and Wang [21] regarding the Zariski closure of polynomials over finite fields, thus resolving the question of Chen et al.

1.4 The use of our results in subsequent work

After a preliminary version of our results appeared in [19], our results, specifically the polynomial approximations to $\mathsf{AC}^0[\oplus]$ in the low-error regime, have been used in the following other results. Williams [33] uses it to design faster algorithms to solve 0-1 integer linear programs. Oliveira and Santhanam [22] use a variant over the field $\mathbb{F}_p$ to prove lower bounds for $\mathsf{AC}^0[p]$ *compression protocols*[2] for the Majority function. Recently, Harsha and the second author [17] used a variant over the reals to prove that $(\log s)^{O(d)} \cdot \log(1/\varepsilon)$-wise independence ε-fools AC^0 circuits of size s and depth d, improving upon results of Braverman [8] and Tal [30].

2 Preliminaries

We begin by formally defining certifying polynomials. Throughout the paper, we identify $\{0,1\}$ with $\mathbb{F}_2$. Given a function $f : \{0,1\}^n \to \{0,1\}$, we use $\mathrm{Supp}(f)$ to denote the set $f^{-1}(1)$.

Definition 2.1 (Certifying polynomial)**.** A *non-zero multilinear* polynomial

$$P(X_1,\ldots,X_n) \in \mathbb{F}_2[X_1,\ldots,X_n]$$

is a *certifying polynomial* for a function $f : \{0,1\}^n \to \{0,1\}$ if f is constant on the non-empty set $\mathrm{Supp}(P) \subseteq \mathbb{F}_2^n$.

This definition is very similar to the notions of *weak*-2 *degree* and *algebraic immunity*, that already appear in the literature [15, 3, 9]. It is also similar in spirit to the notion of *weak sign degree* that appears in the paper of Aspnes, Beigel, Furst, and Rudich [7].

[2] This is not the same as the algorithmic compression problem we consider in this paper. We refer the reader to [22] for details regarding these protocols.

We observe the following upper bound on the certifying polynomial degree of any Boolean function.

Lemma 2.2. *Any $f:\{0,1\}^n \to \{0,1\}$ has a certifying polynomial P of degree at most $\lceil n/2 \rceil$.*

Proof. Let $N = \sum_{i=0}^{\lceil n/2 \rceil} \binom{n}{i}$. Since $N > 2^{n-1}$, we see that one among the sets $f^{-1}(0)$ and $f^{-1}(1)$ must have size less than N. Without loss of generality, we assume that $|f^{-1}(0)| \le 2^{n-1}$. Consider the problem of finding a non-zero polynomial P of degree at most $\lceil n/2 \rceil$ that vanishes over $f^{-1}(0)$. Finding the coefficients of P amounts to finding a non-zero solution to a homogeneous system of linear equations with N variables and $|f^{-1}(0)| < N$ constraints. By standard linear algebra, we know that there is a P as required. □

Lemma 2.2 is tight, for example, for the Majority function on n variables when n is odd (see Section 4.2).

We now define Approximate Majority.

Definition 2.3 (Approximate Majority)**.** An $(a, n-a)$-*Approximate Majority* is a Boolean function $f:\{0,1\}^n \to \{0,1\}$ such that the following holds.

- $f(x) = 0$ for every x of Hamming weight at most a.
- $f(x) = 1$ for every x of Hamming weight at least $n-a$.

If we omit the $(a, n-a)$, we assume $a = n/4$.

We assume standard notions about circuit classes $\mathsf{AC}^0, \mathsf{AC}^0[\oplus], \mathsf{AC}^0[p]$ for prime p etc. (see, e. g., [6]). The *size* of a circuit will always mean the number of *gates* in the circuit.[3]

Ajtai and Ben-Or [2] showed that for $a \le n/2 - n/(\log n)^{O(1)}$, there exists an $(a, n-a)$-Approximate Majority computable in AC^0. We will use a uniform and more general version of this result, due to Ajtai [1].

Theorem 2.4 (Ajtai [1])**.** *For any $n \in \mathbb{N}$, $\delta \in (0, 1/2)$ and depth $d \ge 3$, there exist $((1/2-\delta)n, (1/2+\delta)n)$-Approximate Majorities computable by uniform AC^0 circuits of size $2^{(1/\delta)^{O(1/d)}} \cdot n^{O(1)}$ and depth d.*

We also use the well-studied notion of approximating polynomials, introduced by Razborov [24].

Definition 2.5 (ε-error approximating polynomial)**.** An ε*-error approximating polynomial* for a function f is a polynomial P such that $\Pr_{x \in \{0,1\}^n}[f(x) = P(x)] \ge 1 - \varepsilon$.

We recall the notion of the *Compression problem* for a circuit class $\mathcal{C}$ from the result of Chen et al. [12]. The input to the problem is the truth table of a Boolean function $f:\{0,1\}^n \to \{0,1\}$ which is promised to have a "small" circuit from the class $\mathcal{C}$. The desired output is a general Boolean circuit C (not necessarily from the class $\mathcal{C}$) of small size that computes the function f; the size of C should be smaller than the trivial $2^n/n$ that is achievable for any Boolean function. Moreover, we require the algorithm that constructs C to run in time polynomial in the size of its input, which is in time $\mathrm{poly}(2^n)$.

[3] There are two standard notions of size for a circuit, corresponding to the number of gates and to the number of wires in the circuit. While these quantities are polynomially related, it matters which one we are dealing with in the near-linear regime, as in the present paper. The lower bound on the number of gates we prove here implies a similar lower bound for the number of wires.

The aforementioned paper of Chen et al. [12] that introduced this problem showed that there is a polynomial-time compression algorithm for AC^0 in the following sense: given as input the truth table of $f : \{0,1\}^n \to \{0,1\}$ which has an AC^0 circuit of size s and depth $d = O(1)$, the algorithm outputs a circuit computing f of size at most $2^{n-n/(O(\log s))^{d-1}}$. Similar compression algorithms were also obtained for functions that have de Morgan formulas of size at most $n^{2.5-\Omega(1)}$, Boolean formulas (over the complete basis) of size $n^{2-\Omega(1)}$ and read-once branching programs of size $2^{n(1/2-\Omega(1))}$. We refer the reader to [12] for the compression obtained in these cases.

3 Results

Our main lower bound result is the following.

Theorem 3.1. *For every constant $d \in \mathbb{N}$, there is an $\varepsilon_d > 0$ such that any depth-d $\mathsf{AC}^0[\oplus]$ circuit that computes an Approximate Majority must have size $\Omega(n^{1+\varepsilon_d})$.*

By Theorem 2.4, this implies that uniform AC^0 circuits of polynomial size can be more powerful than linear-sized non-uniform $\mathsf{AC}^0[\oplus]$ circuits.

The proof of Theorem 3.1 yields $\varepsilon_d = 1/2^{d+1}$. This is marginally better than the lower bound of $\Omega(n^{1+1/4^d})$ obtained by Chaudhuri and Radhakrishnan [11] for the case of AC^0. (The improvement is due to a slightly different deterministic restriction method: whereas [11] try to remove both high fan-in and high fan-out gates from the circuit, we handle only the high fan-in gates.) Also, as already showed by [11], this lower bound cannot be substantially improved since Approximate Majorities can be computed by AC^0 circuits of depth d and size $n^{1+1/2^{\Omega(d)}}$.

The proof of Theorem 3.1 follows from the two lemmas below. The first one states that every function with a near-linear $\mathsf{AC}^0[\oplus]$ circuit has a certifying polynomial of low degree. The next one states that an Approximate Majority cannot have this property. The proofs of these lemmas appear in Section 4.

Lemma 3.2 (Near-linear-size $\mathsf{AC}^0[\oplus]$ circuits have low-degree certifying polynomials)**.** *For every constant $d \in \mathbb{N}$, there is an $\varepsilon_d > 0$ such that for every depth-d $\mathsf{AC}^0[\oplus]$ circuit C of size $s \leq n^{1+\varepsilon_d}$, C has a certifying polynomial of degree $o(n)$.*

Lemma 3.3 (Approximate Majority does not have any low-degree certifying polynomials)**.** *For every $(a, n-a)$-Approximate Majority f, there do not exist any certifying polynomials for f of degree $\leq a$.*

Next we state our results on certifying polynomials for general $\mathsf{AC}^0[\oplus]$ circuits. This result should be contrasted with the fact that every function has a certifying polynomial of degree at most $\lceil n/2 \rceil$ (Lemma 2.2).

Theorem 3.4. *For every $s > 0$ and constant $d > 0$, every $\mathsf{AC}^0[\oplus]$ circuit C of size s and depth d has a certifying polynomial of degree at most $n/2 - n/(\log s)^{\Theta(d)}$.*

We also show that this is essentially tight.

Lemma 3.5. *For every $s > n^{\Omega(1)}$, there exist $\mathsf{AC}^0[\oplus]$ circuits C on n input bits with size s, such that every certifying polynomial for C has degree at least $n/2 - n/(\log s)^{\Theta(d)}$.*

These results are proved in Section 5. The main ingredient in the proof of Theorem 3.4 is the following strengthening of Razborov's original theorem on approximating polynomials.

Lemma 3.6. *There is an absolute constant $c > 0$ such that the following holds. For any $\varepsilon \in (0, 1/2)$, any $\mathsf{AC}^0[\oplus]$ circuit C of size s and depth d has an ε-error approximating polynomial of degree at most $(c \log s)^{d-1} \cdot (\log(1/\varepsilon))$.*

We also show in Section 5 how Theorem 3.4 gives an alternate proof of Razborov's fundamental result that Majority does not have subexponential-size $\mathsf{AC}^0[\oplus]$ circuits.

We also prove lower bounds for the degree of approximating polynomials for $\mathsf{AC}^0[\oplus]$ circuits, showing the near-tightness of Lemma 3.6. The proof appears in Section 6.

Theorem 3.7. *For every $s, \varepsilon > 0$, and every constant $d \geq 3$, there exist $\mathsf{AC}^0[\oplus]$ circuits C of size s and depth d such that for every polynomial P which is an ε-error approximating polynomial for C, we have*

$$\deg(P) \geq \left(\log s - O\left(\log\log\frac{1}{\varepsilon}\right)\right)^{\Theta(d)} \cdot \log\frac{1}{\varepsilon}\,.$$

Finally, we use certifying polynomials to give compression algorithms for $\mathsf{AC}^0[\oplus]$.

Theorem 3.8. *There is a polynomial-time algorithm which, when given as input the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ and parameters s and $d = O(1)$ such that f has an $\mathsf{AC}^0[\oplus]$ circuit of size s and depth d, outputs a circuit C of size $2^{n-n/(O(\log s))^{2(d-1)}}$ computing f.*

4 Superlinear $\mathsf{AC}^0[\oplus]$ lower bounds for computing Approximate Majority

In this section, we prove Lemma 3.2 and Lemma 3.3, thus completing the proof of Theorem 3.1.

4.1 Near-linear-size $\mathsf{AC}^0[\oplus]$ circuits have low-degree certifying polynomials

We now prove Lemma 3.2, restated below.

Lemma 3.2. *(Restated from Section 3.) For every constant $d \in \mathbb{N}$, there is an $\varepsilon_d > 0$ such that for every depth-d $\mathsf{AC}^0[\oplus]$ circuit C of size $s \leq n^{1+\varepsilon_d}$, C has a certifying polynomial of degree $o(n)$.*

We will actually construct a polynomial P such that $P^{-1}(0)$ is non-empty and C is constant on $P^{-1}(0)$. The polynomial $P' = 1 - P$ is then a certifying polynomial for C. (Recall that we are working over the field $\mathbb{F}_2$.)

It will also be more convenient to work with a system of polynomials as opposed to a single polynomial. Given a feasible system of polynomial equations in n variables, $X_1, X_2, \ldots, X_n$, say

$$\begin{aligned} p_1(X) &= 0\,,\\ p_2(X) &= 0\,,\\ &\vdots\\ p_t(X) &= 0\,, \end{aligned}$$

we define the degree of the system to be $\sum_{i=1}^t \deg(p_i)$. The set of solutions to this system is exactly the set of roots of $1-\prod_{i=1}^t(1-p_i)$, which is a polynomial of degree at most $\sum_{i=1}^t \deg(p_i)$.

Given a feasible system $\mathcal{P}$ of polynomial equations, we denote by $\mathrm{Sol}(\mathcal{P})$ the non-empty set of solutions of $\mathcal{P}$. When $\mathcal{P}$ consists of a single polynomial p, we use $\mathrm{Sol}(p)$ instead of $\mathrm{Sol}(\mathcal{P})$. By a *restriction*, we will mean simply a feasible system of polynomial equations.

Given a restriction $\mathcal{P}$ and a Boolean circuit C, we will denote by $C|_{\mathcal{P}}$ the circuit C restricted to inputs from $\mathrm{Sol}(\mathcal{P})$. We say a gate g of the circuit C is *live* under the restriction given by $\mathcal{P}$ if g takes values 0 as well as 1 under inputs from $\mathrm{Sol}(\mathcal{P})$. Note that if a gate g is not live under a restriction, we can simplify the circuit C to a smaller circuit C' which computes the same function on the restricted inputs.

We say that a circuit C is *live* under the restriction $\mathcal{P}$ if every gate of C is live under $\mathcal{P}$. The above implies that, given any circuit C and restriction $\mathcal{P}$, there exists a live circuit C' of size at most the size of C that computes the same function as C on inputs from $\mathrm{Sol}(\mathcal{P})$.

Proof of Lemma 3.2. The proof will proceed as follows. After restricting the given circuit C to the roots of a well-chosen low-degree polynomial restriction $\mathcal{P}$, we will obtain an equivalent circuit C' that has the property that each of the AND and OR gates of C' have very small fan-in (say n^{ε} for $\varepsilon \ll 1/d$). At this point, the entire circuit C' computes a low-degree polynomial p and by setting p to a feasible value, we finish the proof of the lemma.

Say we have an increasing sequence of numbers $1 < D_1 < \cdots < D_d$. (We will fix the exact values of D_i $(i \geq 1)$ later.) We wish to obtain a restriction $\mathcal{P}$ under which C is equivalent to a circuit C' which has the property that every AND and OR gate at height i has fan-in at most D_i. In particular, the circuit C' is a function of at most $D_1D_2\cdots D_d$ variables and hence is a polynomial of degree at most $D_1D_2\cdots D_d$.

We proceed to construct a suitable restriction $\mathcal{P}$ in d steps. After the i-th step, we obtain a restriction $\mathcal{P}_i$ under which there is a circuit C_i of size at most s for which the above fan-in bound holds for all heights $j \leq i$. Assuming that the $(i-1)$-st step has been completed, we describe how Step i is performed for $i \geq 1$. (Note that nothing needs to be done for height 0.)

We assume that C_i is live. Otherwise, we can obtain and work with an equivalent circuit that is of at most the size of C_i and satisfies the same fan-in restrictions as C_i. Let B_i denote the "bad" gates at height i, that is, the AND and OR gates at height i that have fan-in at least D_i. We use a basic subroutine $\mathrm{Fix}(i,C_i)$ that simplifies the circuit C_i by augmenting the restriction $\mathcal{P}_i$ as follows.

The subroutine $\mathrm{Fix}(i,C_i)$. Since there are at least $|B_i|D_i$ wires between gates in B_i and lower levels (which contain at most s gates), there is some gate g at height less than i that is adjacent to $|B_i|D_i/s$ gates. By the fan-in restrictions on C_i, this gate computes a polynomial p_g of degree at most $D_1\cdots D_{i-1}$. (The empty product in the case $i=1$ is assumed to be 1.) Moreover, since the circuit C_i is live, this gate can be set to both 0 and 1. We wish to add the restriction $p_g = 0$ or $p_g - 1 = 0$ to $\mathcal{P}_i$ corresponding to setting the gate to 0 or 1 respectively. Setting the gate g to 1 sets all the OR gates that g feeds into to 1 and setting g to 0 sets all the AND gates that g feeds into to 0. Hence, there is some setting that sets at least $|B_i|D_i/2s$ many gates in B_i to constant. We set the gate g to this Boolean value.

Note that $\mathrm{Fix}(i,C_i)$ reduces the number of live bad gates to at most $|B_i|(1-D_i/2s)$. We are now ready to describe Step i. Until the set of bad gates B_i is empty, we repeatedly call the subroutine $\mathrm{Fix}(i,C_i')$ where C_i' represents the circuit we currently have. After an application of the subroutine $\mathrm{Fix}(i,C_i')$ adds another equation to our current restriction $\mathcal{P}_i'$, we fix the non-live gates and simplify the circuit until it becomes live again. (This process, of course, does not increase the fan-in of any gate.) Note that since we are only setting live gates, the system of polynomial equations $\mathcal{P}'_i$ we maintain is feasible. Moreover, since the size of B_i is falling by a factor $\beta \leq (1 - D_i/2s) \in (0,1)$ after each application of $\mathrm{Fix}(i, C'_i)$ and $|B_i| \leq s \leq n^{O(1)}$, we need to apply $\mathrm{Fix}(i, C'_i)$ at most

$$\frac{2s\log|B_i|}{D_i} = O(s\log n/D_i)$$

times to reduce B_i to the empty set.

Let us analyze the total degree of the equations added to the restriction during the i-th step. Each equation added is a polynomial of degree at most $D_1 D_2 \cdots D_{i-1}$. Hence, the total degree of the added equations is $O(s \log n D_1 D_2 \cdots D_{i-1}/D_i)$.

At the end of Step d, we have a circuit C_d computing a polynomial of degree at most $D' = D_1 D_2 \cdots D_d$ that agrees with the original circuit C on a restriction of degree at most

$$D'' = O(s\log n)\left(\frac{1}{D_1} + \frac{D_1}{D_2} + \frac{D_1 D_2}{D_3} + \cdots + \frac{D_1 D_2 \cdots D_{d-1}}{D_d}\right).$$

We would like to set $D_1, \ldots, D_d$ such that both D' and D'' to be $o(n)$. We will choose K and the D_i such that $D_1 D_2 \cdots D_{i-1}/D_i = K^{-1}$ for each i. This implies that $D_i = K^{2^{i-1}}$. Furthermore, we have $D' \leq K^{2^d}$ and $D'' \leq O(s \log n/K)$.

Setting $K = n^{1/(2^d+1)}$ and $\varepsilon_d = 1/2^{d+1}$, we get D' as well as D'' are $o(n)$ as long as $s \leq n^{1+\varepsilon_d}$. Thus, by setting the polynomial p computed by the circuit C_d to some feasible value, we obtain a restriction of degree $D' + D'' = o(n)$ under which the circuit C becomes constant.

As mentioned above, this implies that there is a certifying polynomial for C of degree $o(n)$. $\square$

4.2 Approximate Majority does not have any low-degree certifying polynomials

Lemma 3.3. *(Restated from Section 3.) For every $(a, n-a)$-Approximate Majority f, there do not exist any certifying polynomials for f of degree $\leq a$.*

Proof. Let $p \in \mathbb{F}_2[X_1, \ldots, X_n]$ be any non-zero multilinear polynomial of degree $d \leq a$. We will show that it cannot be that f is constant on $\mathrm{Supp}(p)$.

Our intermediate claim is that $\mathrm{Supp}(p)$ intersects every Hamming ball of radius a. By translating p if necessary, we may assume that the Hamming ball is centered at the origin, and thus we seek to prove that there is a point of Hamming weight at most a where p does not vanish.

Given the intermediate claim, it follows that there exist $x_0, x_1 \in \mathbb{F}_2^n$ with $p(x_0)$ and $p(x_1)$ both non-zero such that the Hamming weight of x_0 is at most a, and the Hamming weight of x_1 is at least $n-a$. Thus f cannot be constant on $\mathrm{Supp}(p)$.

Now we prove the claim. Notice that since $\mathrm{Supp}(p)$ is nonempty, it is a non-zero polynomial. Assume that

$$p(X_1, \ldots, X_n) = \sum_{S \subseteq [n] : |S| \leq d} \alpha_S \prod_{i \in S} X_i$$

where $\alpha_S \in \mathbb{F}_2$ for each S.

Fix $R \subseteq [n]$ such that $\alpha_R \neq 0$ but $\alpha_S = 0$ for all $S \subsetneq R$. Let $x \in \mathbb{F}_2^n$ be the input with 1s at exactly the indices in R. By our choice of R, for every monomial $\prod_{i \in S} X_i$ of degree at most d, either $S = R$ and hence $\prod_{i \in S} x_i = 1$; or $\alpha_S = 0$; or $S \nsubseteq R$ and hence $\prod_{i \in S} x_i = 0$. Thus, we get that $p(x) = \alpha_R \neq 0$. Hence $x \in \mathrm{Supp}(p)$.

Moreover, the Hamming weight of x is equal to the size of S which is at most $\deg(p) \leq a$. Hence, we see that $\mathrm{Supp}(p)$ does intersect the Hamming ball of radius a. This completes the proof of the claim, and hence the proof of Lemma 3.3. □

Remark 4.1. In particular, Lemma 3.3 implies that for odd n, any certifying polynomial for the Majority function on n variables must have degree at least $\lceil n/2 \rceil$, matching the bound given for arbitrary functions in Lemma 2.2.

5 Certifying polynomials for general $\mathsf{AC}^0[\oplus]$ circuits

Given the results of the previous section, it makes sense to ask what are the lowest degree certifying polynomials we can obtain for general (i. e., significantly larger than linear-sized) $\mathsf{AC}^0[\oplus]$ circuits. By Lemma 2.2, we know that *every* function, irrespective of its complexity, has a certifying polynomial of degree at most $n/2$ and as pointed out in Remark 4.1, this bound is tight for general functions. In this section, we use Razborov's approximations for $\mathsf{AC}^0[\oplus]$ circuits by probabilistic polynomials to derive somewhat better certifying polynomials for functions with small $\mathsf{AC}^0[\oplus]$ circuits. In particular, we show that polynomial-sized $\mathsf{AC}^0[\oplus]$ circuits have certifying polynomials of degree $n/2 - n/(\log n)^{O(1)}$.

Though this improvement over the trivial $n/2$ bound might seem small, the existence of such certifying polynomials is quite powerful. We demonstrate this by showing how this fact, along with Lemma 3.3, can be used to give a (slightly) conceptually different proof of Razborov's result that Majority does not have subexponential-size $\mathsf{AC}^0[\oplus]$ circuits. We note that the proof is essentially unchanged at a technical level from the proofs of [24, 28], but the higher-order concepts involved seem curiously different. More specifically, this proof is more reminiscent of the result of Aspnes et al. [7] in the context of AC^0 augmented with a single threshold gate on top.

The main theorem of this section is the following.

Theorem 3.4. *(Restated from Section 3.) For every $s > 0$ and constant $d > 0$, every $\mathsf{AC}^0[\oplus]$ circuit C of size s and depth d has a certifying polynomial of degree at most $n/2 - n/(\log s)^{\Theta(d)}$.*

Theorem 3.4 shows that functions computed by small subexponential-size $\mathsf{AC}^0[\oplus]$ circuits have certifying polynomials of non-trivially small degree.

We will need to use probabilistic polynomials in the proof.

Definition 5.1 (Probabilistic polynomials)**.** An *ε-error probabilistic polynomial* of degree D for a function $f : \{0,1\}^n \to \{0,1\}$ is a random polynomial $\mathbf{P}$ of degree at most D (chosen according to some distribution over polynomials of degree at most D) such that for any $x \in \{0,1\}^n$, we have $\Pr_{\mathbf{P}}[f(x) = \mathbf{P}(x)] \geq 1 - \varepsilon$.

Clearly, if a function f has an ε-error probabilistic polynomial $\mathbf{P}$ of degree D, then by the probabilistic method, it has an ε-error approximating polynomial P of degree D as well.

We need the following well-known theorem, due to Razborov, on the existence of ε-error probabilistic polynomials for $\mathsf{AC}^0[\oplus]$. A gate of a circuit C is said to be *internal* if it is not a leaf.

Theorem 5.2 (Razborov [24])**.** *For any $\varepsilon \in (0,1/2)$, any $\mathsf{AC}^0[\oplus]$ circuit C with at most s internal gates and depth $d \geq 1$ has an ε-error probabilistic polynomial of degree at most $(\log(s/\varepsilon))^d$. In particular, C has an ε-error approximating polynomial of degree at most $(\log(s/\varepsilon))^d$.*

Using Theorem 5.2 directly in our arguments would only give us a version of Theorem 3.4 with weaker parameters. To obtain the parameters of Theorem 3.4, we need a strengthening of Theorem 5.2 that does better for small ε. The proof follows quite simply from Razborov's cited result, although to the best of our knowledge, this has not been observed in the literature.

Lemma 3.6. *(Restated from Section 3 in a stronger form.) The following holds for the some absolute constant $c > 0$. For any $\varepsilon \in (0,1/2)$, any $\mathsf{AC}^0[\oplus]$ circuit C of size s and depth d has an ε-error probabilistic polynomial of degree at most $(c\log s)^{d-1} \cdot (\log(1/\varepsilon))$ for some absolute constant $c > 0$. In particular, C has an ε-error approximating polynomial of degree at most $(c\log s)^{d-1} \cdot (\log(1/\varepsilon))$.*

Proof. Let C be an $\mathsf{AC}^0[\oplus]$ circuit of size s and depth d. Let g be the output gate of the circuit and let $C_1,\ldots,C_k$ ($k \leq s$) be the depth-$(d-1)$ subcircuits of C feeding into g. By Theorem 5.2, we know that each C_i ($i \in [k]$) has a $(1/10s)$-error approximating polynomial $\mathbf{P}_i$ of degree at most $(O(\log s))^{d-1}$. Also by Theorem 5.2, we know that the function computed by g has an $\mathsf{AC}^0[\oplus]$ circuit with just one *internal* gate and hence has a $(1/10)$-error approximating polynomial $\mathbf{P}$ of degree $O(1)$. The probabilistic polynomial $\mathbf{P}' := \mathbf{P}(\mathbf{P}_1,\ldots,\mathbf{P}_k)$ is a $1/5$-error probabilistic polynomial for C, since for any $x \in \{0,1\}^n$,

$$\begin{aligned}
\Pr_{\mathbf{P}'}[C(x) \neq \mathbf{P}'(x)] &\leq \Pr_{\mathbf{P}_1,\ldots,\mathbf{P}_k}[\exists i \in [k] :\ C_i(x) \neq \mathbf{P}_i(x)] + \Pr_{\mathbf{P}}[g(C_1(x),\ldots,C_k(x)) \neq \mathbf{P}(C_1(x),\ldots,C_k(x))] \\
&\leq \sum_{i\in[k]} \Pr_{\mathbf{P}_i}[C_i(x) \neq \mathbf{P}_i(x)] + \Pr_{\mathbf{P}}[g(C_1(x),\ldots,C_k(x)) \neq \mathbf{P}(C_1(x),\ldots,C_k(x))] \\
&\leq k/10s + 1/10 \leq 1/10 + 1/10 = 1/5\,.
\end{aligned}$$

Note that $\mathbf{P}'$ has degree at most $(O(\log s))^{d-1}$. Let $\ell = c'\log(1/\varepsilon)$ for a constant c' that we will choose later in the proof. Let $\mathbf{P}'_1,\ldots,\mathbf{P}'_\ell$ be ℓ independent copies of the probabilistic polynomial $\mathbf{P}'$. Let $\mathbf{Q}$ denote the probabilistic polynomial $\mathrm{Maj}(\mathbf{P}'_1,\ldots,\mathbf{P}'_\ell)$, where Maj is just the polynomial of degree at most ℓ that computes the majority of ℓ bits. Note that $\mathbf{Q}$ is of degree at most

$$\deg(\mathrm{Maj}) \cdot \max_i \deg(\mathbf{P}'_i) = (O(\log s))^{d-1} \cdot \ell = (O(\log s))^{d-1} \cdot \log(1/\varepsilon)\,.$$

We claim that $\mathbf{Q}$ is an ε-error probabilistic polynomial for C, which will finish the proof of the corollary.

For any input $x \in \{0,1\}^n$, each $\mathbf{P}'_j(x)$ predicts the value of $C(x)$ correctly with probability $4/5$. Now, for $\mathbf{Q}(x)$ to predict $C(x)$ incorrectly, a *majority* of the $\mathbf{P}'_j$ ($j \in [\ell]$) must predict the value of $C(x)$ incorrectly and by a Chernoff bound, the probability of this is bounded by $\exp(-\Omega(\ell))$, which is at most ε for a large enough constant $c' > 0$. □

The next lemma shows that functions with low-degree ε-error approximating polynomials also have low-degree certifying polynomials.

Lemma 5.3. *Suppose $f : \{0,1\}^n \to \{0,1\}$ has a degree D ε-error approximating polynomial. Then f has a certifying polynomial of degree at most*

$$\frac{n}{2} - c_1\sqrt{n\log\frac{1}{\varepsilon}} + D\,,$$

where c_1 is an absolute constant.

Proof. Let P be the given ε-error approximating polynomial. Let S be the set of points where P differs from f. We have $|S| \le \varepsilon \cdot 2^n$.

Let D_0 be the smallest integer such that

$$\binom{n}{0} + \binom{n}{1} + \cdots + \binom{n}{D_0} > |S|\,.$$

By linear algebra, there is a non-zero polynomial Q of degree at most D_0 that vanishes on S. Note that one of $Q \cdot P$ and $Q \cdot (1-P)$ is a non-zero polynomial. Moreover, for any input $x \in \mathrm{Supp}(Q)$, we have $P(x) = f(x)$.

Thus, it follows that one of $Q \cdot P$ or $Q \cdot (1-P)$ is a certifying polynomial for f with degree at most $D_0 + D$ (provided $D_0 + D < n$; if not then the result is vacuously true). To finish the proof, we note that by standard binomial estimates,

$$D_0 \le \frac{n}{2} - c_1\sqrt{n\log\frac{1}{\varepsilon}}\,.$$

(See, e. g., [22, Section B, Lemma 2.4].) □

Proof of Theorem 3.4. Combining Lemma 3.6 and Lemma 5.3, we conclude that for every $\varepsilon > 0$. C has a certifying polynomial of degree at most

$$\frac{n}{2} - c_1\sqrt{n \cdot \log\frac{1}{\varepsilon}} + (c_2 \log s)^{d-1} \cdot \log(1/\varepsilon)\,,$$

where $c_1, c_2 > 0$ are absolute constants. In particular, setting $\varepsilon = \exp(-n/(\log s)^{\Theta(d)})$, we get that C has a certifying polynomial of degree at most $n/2 - n/(\log s)^{\Theta(d)}$. □

Combining Theorem 3.4 with Lemma 3.3 (and using the fact that Majority is an $(n/2-1, n/2+1)$-Approximate Majority), we get an alternate proof of the fact that Majority cannot be computed by $\mathsf{AC}^0[\oplus]$ circuits of size smaller than $\exp(n^{\Omega(1/d)})$.

Finally, we show that the bound of Theorem 3.4 is essentially tight.

Lemma 5.4. *(Restated from Section 3.) For every $s > n^{\Omega(1)}$, there exist $\mathsf{AC}^0[\oplus]$ circuits C on n input bits with size s, such that every certifying polynomial for C has degree at least $n/2 - n/(\log s)^{\Theta(d)}$.*

Proof. Let δ be a parameter to be specified later. Let C be the $\mathsf{AC}^0[\oplus]$ circuit for $((1/2-\delta)n, (1/2+\delta)n)$-Approximate Majority given by Theorem 2.4. Then we have

$$|C| = 2^{(1/\delta)^{O(1/d)}} \cdot n^{O(1)}\,.$$

We choose δ so that $|C| = s$; this gives $\delta = 1/(\log s - c\log n)^{\Omega(d)}$.

By Lemma 3.3, any certifying polynomial for C has degree at least

$$n \cdot \left(\frac{1}{2} - \delta\right) = \frac{n}{2} - \frac{n}{(\log s)^{\Theta(d)}}. \qquad \square$$

6 Lower bounds for approximating polynomials

We now use the tools of the previous two sections to show near-optimal lower bounds on the degree of approximating polynomials for $\mathsf{AC}^0[\oplus]$ circuits. It is a folklore fact that ε-error approximations for $\mathsf{AC}^0[\oplus]$ circuits of size s and depth d are required to have degree at least $\max\{(\log s)^{\Omega(d)}, \log(1/\varepsilon)\}$. In this section, we show a stronger lower bound of $\Theta((\log s)^{\Omega(d)} \cdot \log(1/\varepsilon))$, which essentially matches the upper bound obtained in Lemma 3.6. Our lower bound example is just a suitable Approximate Majority and thus holds even for AC^0 circuits.

We prove the lower bound by exploiting Lemma 5.3 in the contrapositive. Since there are Approximate Majorities that are efficiently computable in AC^0, by Lemma 3.3, we know that AC^0 circuits can compute functions that do not have efficient certifying polynomials. We can then use Lemma 5.3 to infer a lower bound on the degree of ε-error approximations to AC^0 circuits.

Theorem 3.7. *(Restated from Section 3.) For every $s, \varepsilon > 0$, and every constant $d \geq 3$, there exist $\mathsf{AC}^0[\oplus]$ circuits C of size s and depth d such that for every polynomial P which is an ε-error approximating polynomial for C, we have*

$$\deg(P) \geq \left(\log s - O\left(\log\log\frac{1}{\varepsilon}\right)\right)^{\Theta(d)} \cdot \log\frac{1}{\varepsilon}.$$

Proof. Let δ and m be parameters (to be specified later). Let C be an $\mathsf{AC}^0[\oplus]$ circuit on m inputs which computes a $((1/2-\delta)m, (1/2+\delta)m)$-Approximate Majority. By Theorem 2.4, such an AC^0 circuit can be taken to have depth d and size at most $2^{(1/\delta)^{O(1/d)}} \cdot m^{O(1)}$. We will choose m and δ so that this size equals s.

Suppose P is an ε-error approximating polynomial for C with degree D. By Lemma 5.3, there is a degree

$$\frac{m}{2} - c_1\sqrt{m\log\frac{1}{\varepsilon}} + D$$

polynomial Q which is a certifying polynomial for C.

But since C is a $((1/2-\delta)m, (1/2+\delta)m)$-Approximate Majority, Lemma 3.3 tells us that that $\deg(Q) \geq (1/2-\delta)\cdot m$.

Putting this together, we get that

$$D \geq c_1\sqrt{m\log\frac{1}{\varepsilon}} - \delta\cdot m.$$

We now choose m, δ suitably so that we get the claimed lower bound on D in terms of s and ε. We set m, δ so that

$$c_1\sqrt{m\log\frac{1}{\varepsilon}} = 2\delta\cdot m, \tag{6.1}$$

and so that s is the size of the circuit C. That is,

$$s = \underbrace{2^{(1/\delta)^{O(\frac{1}{d})}}}_{A} \cdot \underbrace{m^{O(1)}}_{B}. \tag{6.2}$$

Note that from (6.1), we get

$$m = \Theta\left(\frac{\log \frac{1}{\varepsilon}}{\delta^2}\right). \tag{6.3}$$

We claim that

$$m \geq \left(\log s - c_2 \log\log \frac{1}{\varepsilon}\right)^{\Theta(d)} \cdot \log \frac{1}{\varepsilon}. \tag{6.4}$$

for a suitably large absolute constant $c_2 > 0$ defined later. Observe that this bound is trivial if $\log s \leq c_2 \log\log 1/\varepsilon$ and thus we may assume that $c_2 \log\log 1/\varepsilon < \log s$.

To prove (6.4), note that if $s \leq A^2$, then we have $1/\delta = (\log s)^{\Omega(d)}$ and in this case (6.4) follows from (6.3).

Otherwise, we see that $s \leq B^2 \leq m^{O(1)}$. In this case, note that if $\log(1/\varepsilon) \leq \sqrt{m}$, then by (6.3), we have $1/\delta = m^{\Omega(1)}$ and hence by (6.2), $s \geq 2^{m^{\Omega(1/d)}}$ which contradicts our upper bound of $s \leq m^{O(1)}$. Hence, we must have $\log(1/\varepsilon) > \sqrt{m}$.

Since $s \leq m^{O(1)}$ and $\log(1/\varepsilon) > \sqrt{m}$, we must have $\log(1/\varepsilon) > s^{\Omega(1)}$, which contradicts the assumption that $c_2 \log\log 1/\varepsilon < \log s$ for a large enough constant c_2.

We have thus proved (6.4). From the lower bound on D obtained above, we get

$$D \geq c_1 \sqrt{m \log \frac{1}{\varepsilon}} - \delta \cdot m \geq \frac{c_1 \sqrt{m \log \frac{1}{\varepsilon}}}{2} \geq \left(\log s - O\left(\log\log \frac{1}{\varepsilon}\right)\right)^{\Theta(d)} \cdot \log \frac{1}{\varepsilon},$$

as desired. □

7 Compression algorithms for $\mathsf{AC}^0[\oplus]$ circuits

In this section, we use certifying polynomials to obtain compression algorithms for $\mathsf{AC}^0[\oplus]$. We devise a deterministic polynomial-time algorithm which, when given as input the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ that has an $\mathsf{AC}^0[\oplus]$ circuit of size s outputs a circuit of size $2^{n-n/\mathrm{polylog}(s)}$ computing f.

Our starting point is the proof of Theorem 3.4 which shows that the input function f has a certifying polynomial of degree at most $D = (n/2) - O(n/(\log s)^{2(d-1)})$. Let $\varepsilon = \exp(-n/(c \log s)^{2(d-1)})$ for a constant $c > 0$ yet to be chosen. To construct such a certifying polynomial, we start with a polynomial P of degree at most $d = (c_2 \log s)^{d-1} \cdot \log(1/\varepsilon)$ that computes f on all but an ε fraction of inputs. Let $\mathcal{E}_P$ be the set of inputs where P does not compute f correctly. We then construct a non-zero polynomial Q of degree $D_0 = (n/2) - c_1 \sqrt{n \log(1/\varepsilon)}$ that vanishes on $\mathcal{E}_P$: in the proof of Theorem 3.4, D_0 is chosen so that the number of monomials of degree at most D_0 is greater than $\varepsilon \cdot 2^n \geq |\mathcal{E}_P|$, which implies that there is a non-zero Q vanishing on $\mathcal{E}_P$; the polynomial $Q \cdot P$ or the polynomial $Q \cdot (1 - P)$ is then a certifying polynomial for f of degree at most $D_0 + d$. Let us further assume that $Q \cdot P$ is non-zero, and is hence a 1-certifying polynomial for f (i. e., $f(x) = 1$ for all x in the support of the certifying polynomial).

Note that this idea also gives us an efficient algorithm for *constructing* such a certifying polynomial. Formally, given the truth table of f, we can efficiently find a certifying polynomial for f of degree at most $D_0 + d$, since the problem of finding a 1-certifying polynomial for f is equivalent to finding a non-zero solution to a system of homogeneous linear equations over $\mathbb{F}_2$ where the variables correspond to coefficients of monomials of degree at most $D_0 + d$.

The foregoing discussion gives us a hint of how to go about compressing the function f. We can try to find a 1-certifying polynomial for f of degree at most $D_0 + d$. Note that (for a suitable choice of c) the number of monomials in such a polynomial is $2^{n-n/(\log s)^{O(d)}}$, and hence this polynomial can be represented as a depth-2 $\mathsf{AC}^0[\oplus]$ circuit of this size (alternately, since the Parity function on m bits has a an circuit over the de Morgan basis of size $O(m)$ [32, Theorem 4.1], we can also represent this polynomial as a circuit over the de Morgan basis of size $2^{n-n/(\log s)^{O(d)}}$). Hence, the certifying polynomial gives us a "small" circuit that computes f correctly on a certain subset of inputs (and in particular is never wrong on inputs of $f^{-1}(0)$).

However, we are looking for a small circuit that computes f *everywhere*. To obtain such a circuit, we try to look for many 1-certifying polynomials $R_1, \ldots, R_m$ and try to cover all the 1-inputs of f. If we are able to do this with a small m, then $\bigvee_{i=1}^m R_i$ computes the function f. There are two things that could go wrong with such an approach.

- Each 1-certifying polynomial R is forced to vanish at all inputs $x \in f^{-1}(0)$. However, this could also force R to vanish at some inputs $y \in f^{-1}(1)$. Such forced inputs y cannot be covered by any 1-certifying polynomial R.
- Each 1-certifying polynomial R that we find might cover very few $y \in f^{-1}(1)$ and hence we might require many 1-certifying polynomials to cover all of $f^{-1}(1)$.

Handling the second of these issues is not too difficult, since we can use a simple linear algebraic argument to show that for each y that is not forced in the above sense, a significant fraction of 1-certifying polynomials cover y. Coupled with a covering argument from [12], we can show that there are a few certifying polynomials that cover all such y.

To get around the first issue, we use a recent result of Nie and Wang [21], which implies that the number of forced y is vanishingly small if the parameters are chosen carefully. We are therefore able to hardcode these y into our circuit without a significant blowup in size. This finishes the proof.

We now state the result of Nie and Wang that we will use. Given a subset $\mathcal{E} \subseteq \{0,1\}^n$ and a parameter $D \le n$, we define the *degree D closure of* $\mathcal{E}$, denoted $\mathrm{cl}_D(\mathcal{E}_P)$, which is the set of all points $y \in \{0,1\}^n$ such that any polynomial Q of degree at most D_1 that vanishes on $\mathcal{E}_P$ vanishes on y.

Theorem 7.1 (Theorem 5.6 in [21]). *Let N_D denote the number of multilinear monomials of degree at most D. Then, we have*

$$\frac{|\mathrm{cl}_D(\mathcal{E})|}{2^n} \le \frac{|\mathcal{E}|}{N_D}.$$

This is actually a slight restatement of the result of Nie and Wang [21], who consider the closure of subsets of $\mathbb{F}^n$ (instead of $\{0,1\}^n$) where $\mathbb{F}$ is an arbitrary finite field. Over the field $\mathbb{F}_2$, this is the same as the result of [21], but it also holds over any field in the form stated above. The proof is a straightforward modification of the proof in [21] and is explicitly observed in [21] in the remarks following the proof of Theorem 5.6. For completeness, we give a proof sketch of the theorem (over any field $\mathbb{F}$) in Appendix A.

We now prove Theorem 3.8.

Theorem 3.8. *(Restated from Section 3.) There is a polynomial-time algorithm which, when given as input the truth table of a function $f : \{0,1\}^n \to \{0,1\}$ and parameters s and $d = O(1)$ such that f has an $\mathsf{AC}^0[\oplus]$ circuit of size s and depth d, outputs a circuit C of size $2^{n-n/(O(\log s))^{2(d-1)}}$ computing f.*

Proof. We assume that d and ε are as above. The constant $c > 0$ in the definition of $\varepsilon > 0$ will be chosen below. We will assume that for the c we choose, the quantity $(c\log s)^{2(d-1)} < n/100$: otherwise, the compression algorithm can just output a trivial circuit of size $2^n/n$ for f.

Let $D_1 = (n/2) - c_3\sqrt{n\log(1/\varepsilon)}$ for a constant $c_3 > 0$ that is chosen to so that the number of monomials of degree at most D_1 is $N_{D_1} \geq \sqrt{\varepsilon}2^n$. (See, e. g., [22, Section B, Lemma 2.4].) We choose c so that $D' = D_1 + d = n/2 - n/(O(\log s))^{(d-1)}$.

We call $y \in f^{-1}(1)$ forced if any polynomial R that vanishes on $f^{-1}(0)$ also vanishes on y. Let $F \subseteq f^{-1}(1)$ be the set of all forced y. We will prove the following two claims.

Claim 7.2. $|F| \leq 2^{n-n/(O(\log s))^{2(d-1)}}$.

Claim 7.3. *There is a polynomial-time algorithm $\mathcal{A}_1$ which when given f, outputs the descriptions of at most $m = O(n)$ 1-certifying polynomials $R_1,\ldots,R_m$ such that for each $y \in f^{-1}(1)\setminus F$, there is an $i \in [m]$ such that $y \in \mathrm{Supp}(R_i)$.*

Given Claims 7.2 and 7.3, the description of the compression algorithm $\mathcal{A}$ is as follows. First run $\mathcal{A}_1$ and obtain a collection of 1-certifying polynomials $R_1,\ldots,R_m$ such that

$$\bigcup_i \mathrm{Supp}(R_i) = f^{-1}(y)\setminus F\,.$$

In particular, if C_i is a circuit of size $2^{n-n/(O(\log s))^{2(d-1)}}$ that accepts exactly the inputs in $\mathrm{Supp}(R_i)$, then $C' = \bigvee_i C_i$ is a circuit of the required size that accepts exactly the set $f^{-1}(y)\setminus F$. The algorithm now constructs a DNF C_F of size $O(n\cdot|F|)$ that accepts exactly the inputs in F. (The set F is easily inferred from the circuit C'.) The circuit C output by the algorithm is $C' \vee C_F$, which computes f by definition and also has the required size.

It remains to prove Claims 7.2 and 7.3, which we do below.

Proof of Claim 7.2. Let P and $\mathcal{E}_P$ be as above. Note that if $y \notin \mathrm{cl}_{D_1}(\mathcal{E}_P)$, then there is a polynomial Q of degree at most D_1 that vanishes at all points in $\mathcal{E}_P$ but not at y. Hence, the polynomial $Q\cdot P$ is a 1-certifying polynomial for f of degree at most D' that is non-zero at y and thus, y is not forced. Stated in the contrapositive, this argument tells us that $F \subseteq \mathrm{cl}_{D_1}(\mathcal{E}_P)$ and therefore, $|F| \leq |\mathrm{cl}_{D_1}(\mathcal{E}_P)|$.

By Theorem 5.6 of Nie and Wang [21], we have

$$\frac{|\mathrm{cl}_{D_1}(\mathcal{E}_P)|}{2^n} \leq \frac{|\mathcal{E}_P|}{N_{D_1}}\,.$$

Since $|\mathcal{E}_P| \leq \varepsilon 2^n$ and $N_{D_1} \geq \sqrt{\varepsilon} 2^n$, we see that the right hand size of the above inequality is bounded by $\sqrt{\varepsilon}$, which implies the claim. $\square$

Proof of Claim 7.3. Let V denote the vector space of polynomials Q of degree at most D' such that Q vanishes on $f^{-1}(0)$. Note that $F' := f^{-1}(1) \setminus F$ satisfies $F' = \bigcup_{Q \in V} \mathrm{Supp}(Q)$. Let $Q_1, \ldots, Q_N$ be a generating set of V. Note that $N \leq 2^n$. A generic element of V is given by $\sum_{i=1}^{N} \alpha_i Q_i$ for some choice of $\alpha_1, \ldots, \alpha_N \in \mathbb{F}_2$; we denote this element by $Q_{\overline{\alpha}}$, where $\overline{\alpha}$ denotes the vector $(\alpha_1, \ldots, \alpha_N)$.

For any $y \in F'$, we have $Q_{\overline{\alpha}}(y) = \sum_i \alpha_i Q_i(y)$, which is a linear function of $\overline{\alpha}$. Since $y \in F'$, it is not forced to 0 and hence not all the $Q_i(y)$ are 0. Thus, for a random choice of the α_i, the probability that $Q_{\overline{\alpha}}(y) \neq 0$ is $1/2$. We can derandomize this argument using binary error-correcting codes.

Say we have vectors $U = \{u_1, \ldots, u_N\} \subseteq \mathbb{F}_2^M$ (where $M = 2^{O(n)}$) that generate an error-correcting code of distance δM for some constant $\delta > 0$. There are many standard constructions of such sets U in time $\mathrm{poly}(2^n)$. (See, e. g., [16, Chapter 9].) Let $\mathcal{M}$ be an $M \times N$ matrix with columns $u_1, \ldots, u_N$. Let $\overline{\alpha}^1, \ldots, \overline{\alpha}^M$ denote the rows of $\mathcal{M}$. For any non-zero $\beta_1, \ldots, \beta_N$ we know that $u = \sum_i \beta_i u_i$ has at least δM many non-zero entries. In other words, for any non-zero vector $\overline{\beta} = (\beta_1, \ldots, \beta_N) \in \mathbb{F}_2^N$ and a random $j \in [M]$, the probability that the inner product of $\overline{\beta}$ and $\overline{\alpha}^j$ is non-zero is at least δ.

We are now ready to describe the algorithm $\mathcal{A}_1$. The algorithm needs to finds $m = O(n)$ elements $R_1, \ldots, R_m$ from V such that $F' \subseteq \bigcup_i \mathrm{Supp}(R_i)$. The algorithm goes through m iterations, the i-th iteration producing a polynomial $R_i \in V$. After each iteration, we ensure that the number of elements in F' left uncovered thus far drops by the constant factor $(1-\delta)$; thus, at the end of $m = 2n/\delta$ iterations, all the elements of F' will be covered.

Let $F_i' = F' \setminus \bigcup_{p<i} \mathrm{Supp}(R_p)$ be the set of elements of F' left uncovered after $i-1$ iterations. In the i-th iteration, the algorithm looks at each of the rows of $\mathcal{M}$ and picks the j such that

$$s_j = \left| \mathrm{Supp}\left(\sum_{i \in [N]} \overline{\alpha}_i^j Q_i \right) \cap F_i' \right|$$

is maximized. We know that $v_y := (Q_1(y), \ldots, Q_N(y))$ is a non-zero vector for any choice of $y \in F_i'$. Hence, for a random $j \in [M]$, the probability that the inner product of $\overline{\alpha}^j$ and v is non-zero is at least δ. By averaging, there must be a $j \in [M]$ such that the inner product of $\overline{\alpha}^j$ and v_y is non-zero for at least a δ-fraction of the $y \in F_i'$. Thus, $|F_{i+1}'| \leq (1-\delta)|F_i'|$. $\square$

$\square$

Remark 7.4. After a manuscript version of this compression algorithm appeared in [29], a beautiful result of Carmosino, Impagliazzo, Kabanets, and Kolokolova [10] gave a different *randomized* compression algorithm for the class of $\mathsf{AC}^0[\oplus]$. The algorithm from [10] is very different from our own and uses in principle only the fact that the lower bounds we have for $\mathsf{AC}^0[\oplus]$ are based on *natural* strategies in the sense of Razborov and Rudich [25]. Our algorithm, though specific to the case of $\mathsf{AC}^0[\oplus]$ (and $\mathsf{AC}^0[p]$, see below), is quantitatively stronger and is in addition deterministic.

8 Extension to the MOD_p case

All the results we have proved extend fairly straightforwardly to the setting of $\mathsf{AC}^0[p]$ circuits where p is a prime. (Some such extensions have already appeared in the literature: Oliveira and Santhanam [22] note that the proof of Lemma 3.6 regarding approximating polynomials extends to the MOD_p setting.) The right definition of certifying polynomials is obtained by simply replacing $\mathbb{F}_2$ by $\mathbb{F}_p$ in Definition 2.1 (where $\mathrm{Supp}(P)$ is the set of points x s. t. $P(x) \neq 0$). Finally, for the Compression algorithm, we need a modification of Theorem 7.1 for the case of $\mathbb{F}_p$. This does not follow immediately from the results in the paper of Nie and Wang [21], since they prove such a result only for the degree D closure over the entire field $\mathbb{F}_p$ and not $\{0,1\}^n \subseteq \mathbb{F}_p^n$. However, as observed by Nie and Wang and also stated in Section 7, a straightforward modification of their argument gives the result for closure over $\{0,1\}^n \subseteq \mathbb{F}^n$, where $\mathbb{F}$ can be any field (possibly even infinite). A proof of this is given in Appendix A.

9 Discussion and open questions

We have seen that certifying polynomials are a natural and useful notion in the context of lower bounds for $\mathsf{AC}^0[\oplus]$ circuits. We also saw that they have a rather interesting interaction with the well-studied notion of approximating polynomials for $\mathsf{AC}^0[\oplus]$ circuits.

The fundamental question we would like to answer is whether we can prove a size-hierarchy theorem for $\mathsf{AC}^0[\oplus]$ analogous to the results of Rossman [26] and Amano [4] for AC^0. It would even be interesting to obtain the weaker separation of uniform $\mathsf{AC}^0[\oplus]$ circuits of size $n^{\log n}$ from polynomial-sized $\mathsf{AC}^0[\oplus]$ circuits. Good candidates for proving these separations seem to be the parity of the number of k-cliques in a graph for the former, and the elementary symmetric polynomial of degree $\log n$ for the latter. We have taken the first step in this direction by demonstrating a function that has polynomial-sized uniform AC^0 circuits but not near-linear sized $\mathsf{AC}^0[\oplus]$ circuits.

It would also be interesting to see whether certifying objects (analogous to the certifying polynomials studied here) exist for other, more powerful, circuit classes, and if they can be used to prove new circuit lower bounds and give new compression algorithms.

10 Acknowledgements

We would like to thank Albert Atserias for asking us about the tradeoff between degree and error for approximating polynomials for $\mathsf{AC}^0[\oplus]$ circuits. We would also like to thank the reviewers of FSTTCS 2012 and Theory of Computing for pointing out some errors and improving the quality of the exposition. The second author would like to thank Abhishek Bhrushundi, Prahladh Harsha, Valentine Kabanets and Antonina Kolokolova for useful discussions.

A Appendix: A version of Theorem 7.1 over other fields

In this section, let $\mathbb{F}$ be an arbitrary field and n any positive integer. Consider the vector space V_n of functions $\{f : \{0,1\}^n \to \mathbb{F}\}$. It is a standard fact that any $f \in V_n$ can be represented uniquely as a multilinear polynomial $P \in \mathbb{F}[X_1, \ldots, X_n]$.

Given $S \subseteq \{0,1\}^n$ and a degree parameter $D \in \mathbb{N}$, we define $\mathrm{cl}_D(S)$ be the set of all $y \in \{0,1\}^n$ such that any multilinear polynomial $P \in \mathbb{F}[X_1, \ldots, X_n]$ of degree at most D that vanishes at all the points in S also vanishes at y.

Our aim is to give a sketch of the following theorem. The theorem closely follows the proof of Nie and Wang [21, Theorem 5.6]. This was already observed for a more general setting in [21] but we give the proof here for the reader's convenience.

Theorem A.1. *Let N_D denote the number of multilinear monomials of degree at most D. Then, we have*

$$\frac{|\mathrm{cl}_D(S)|}{2^n} \leq \frac{|S|}{N_D}.$$

To prove Theorem A.1, we will need some preliminary definitions and facts. Fix any $S \subseteq \{0,1\}^n$ and any multilinear polynomial $P \in \mathbb{F}[X_1, \ldots, X_n]$. We define $P_S : S \to \mathbb{F}$ to be the evaluation vector of P at all the points in S. Let $V_D(S)$ be the vector space $\{P_S \mid P \text{ of degree at most } D\}$. Define $H_D(S)$ to be the dimension of $V_D(S)$. Let $V(S)$ be the vector space of all functions $\{f : S \to \mathbb{F}\}$.

Treating multilinear monomials as sets of variables, we can order them in the graded-lexicographic order. (See, e. g., [13].) That is, $\prod_{i \in A} X_i \leq \prod_{j \in B} X_j$ if either $|A| < |B|$ or if $|A| = |B|$ and the smallest $i \in A \triangle B$ lies in A.

We recall some standard facts. The proofs can be found, e. g., in [28, 13].

Fact A.2.

1. *As vector spaces, $V(S) \cong \mathbb{F}[X_1, \ldots, X_n]/I(S)$ where $I(S)$ is the ideal of all polynomials that vanish at all the points in S.*

2. $\dim(V(S)) = |S|$.

3. *For a set $\mathcal{P} \subseteq \mathbb{F}[X_1, \ldots, X_n]$, let $\mathcal{P}_D$ be the subset of all polynomials in $\mathcal{P}$ of degree at most D. Then, as vector spaces $V_D(S) \cong \mathbb{F}[X_1, \ldots, X_n]_D/I(S)_D$.*

4. *(Macaulay's theorem [20]. Also see [28, Theorem 1].) For a multilinear polynomial P, let* $\mathrm{in}(P)$ *denote its leading term with respect to the graded-lexicographic order. Let* $\mathrm{in}(I(S))$ *denote the set of all multilinear monomials that are leading terms of some polynomial in* $I(S)$ *and* $\mathrm{out}(I(S))$ *denote the space of all multilinear monomials that are* not *members of* $\mathrm{in}(I(S))$. *Then, any* $f \in V(S)$ *can be represented* uniquely *as a linear combination of monomials in* $\mathrm{out}(I(S))$ *and similarly any* $P \in V_D(S)$ *can be represented* uniquely *as a linear combination of monomials in* $\mathrm{out}(I(S))_D$. *In particular,* $|S| = \dim(V(S)) = |\mathrm{out}(S)|$ *and* $H_D(S) = \dim(V(S)_D) = |\mathrm{out}(S)_D|$.

5. $\mathrm{in}(I(S))$ *is an up-set (i. e.,* $\prod_{i \in A} X_i \in \mathrm{in}(I(S))$ *and* $A \subseteq B \subseteq [n]$ *implies that* $\prod_{i \in B} X_i \in \mathrm{in}(I(S))$*). Consequently,* $\mathrm{out}(I(S))$ *is a down-set.*

Finally, we use the following combinatorial lemma that is a simple restatement of Kleitman's lemma [5, Theorem 6.1.3].

Lemma A.3 (Theorem 6.1.3 in [5]). *Let $\mathcal{A},\mathcal{B} \subseteq 2^{[n]}$ be any down-sets. Then*

$$\frac{|\mathcal{A}|}{2^n} \leq \frac{|\mathcal{A}\cap\mathcal{B}|}{|\mathcal{B}|}\,.$$

Lemma A.3 has a nice algebraic corollary.

Corollary A.4. *Fix any $T \subseteq \{0,1\}^n$. Then*

$$\frac{|T|}{2^n} \leq \frac{H_D(T)}{N_D}\,.$$

Proof. We apply Lemma A.3 with $\mathcal{A} = \mathrm{out}(I(T))$ (which is a down-set by Fact A.2) and $\mathcal{B} = \{A \subseteq [n] \mid |A| \leq D\}$. Since $|\mathrm{out}(I(T))| = |T|$ and $|\mathrm{out}(I(T))_D| = H_D(T)$ by Fact A.2, and also $|\mathcal{B}| = \sum_{i=0}^{D}\binom{n}{i} = N_D$, Lemma A.3 immediately implies the statement of the corollary. □

We now prove the main theorem of this section.

Proof of Theorem A.1. Say $T = \mathrm{cl}_D(S)$. Note that $T \supseteq S$.

We know that any polynomial of degree at most D that vanishes on S must also vanish on T. Equivalently, the linear map $\rho : V_D(T) \to V_D(S)$ obtained by restricting each polynomial P_T to points in S has trivial kernel and is hence injective.

Thus, $H_D(T) = \dim(V_D(T)) \leq \dim(V_D(S)) \leq \dim(V(S)) = |S|$. Applying Corollary A.4, we get

$$\frac{|T|}{2^n} \leq \frac{|S|}{N_D}\,,$$

which is the statement of the theorem. □

References

[1] MIKLÓS AJTAI: Approximate counting with uniform constant-depth circuits. In JIN-YI CAI, editor, *Advances in Computational Complexity Theory*, volume 13 of *DIMACS Ser. in Discr. Math. and Theoret. Comput. Sci.*, pp. 1–20. Amer. Math. Soc., 1993. 2, 6

[2] MIKLÓS AJTAI AND MICHAEL BEN-OR: A theorem on probabilistic constant depth computations. In *Proc. 16th STOC*, pp. 471–474. ACM Press, 1984. [doi:10.1145/800057.808715] 2, 6

[3] MICHAEL ALEKHNOVICH AND ALEXANDER A. RAZBOROV: Lower bounds for polynomial calculus: Non-binomial case. In *Proc. 42nd FOCS*, pp. 190–199. IEEE Comp. Soc. Press, 2001. [doi:10.1109/SFCS.2001.959893] 3, 5

[4] KAZUYUKI AMANO: k-subgraph isomorphism on AC^0 circuits. *Comput. Complexity*, 19(2):183–210, 2010. Preliminary version in CCC'09. [doi:10.1007/s00037-010-0288-y] 3, 19

[5] IAN ANDERSON: *Combinatorics of Finite Sets*. Dover, 2011. 20, 21

[6] SANJEEV ARORA AND BOAZ BARAK: *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ACM DL. 6

[7] JAMES ASPNES, RICHARD BEIGEL, MERRICK L. FURST, AND STEVEN RUDICH: The expressive power of voting polynomials. *Combinatorica*, 14(2):135–148, 1994. Preliminary version in STOC'91. [doi:10.1007/BF01215346] 3, 5, 11

[8] MARK BRAVERMAN: Polylogarithmic independence fools AC^0 circuits. *J. ACM*, 57(5):28:1–28:10, 2010. Preliminary version in CCC'09. [doi:10.1145/1754399.1754401] 5

[9] CLAUDE CARLET, DEEPAK KUMAR DALAI, KISHAN CHAND GUPTA, AND SUBHAMOY MAITRA: Algebraic immunity for cryptographically significant Boolean functions: Analysis and construction. *IEEE Trans. Inform. Theory*, 52(7):3105–3121, 2006. [doi:10.1109/TIT.2006.876253] 3, 5

[10] MARCO L. CARMOSINO, RUSSELL IMPAGLIAZZO, VALENTINE KABANETS, AND ANTONINA KOLOKOLOVA: Learning algorithms from natural proofs. In *Proc. 31st Computational Complexity Conf. (CCC'16)*, pp. 10:1–10:24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.10] 18

[11] SHIVA CHAUDHURI AND JAIKUMAR RADHAKRISHNAN: Deterministic restrictions in circuit complexity. In *Proc. 28th STOC*, pp. 30–36. ACM Press, 1996. [doi:10.1145/237814.237824] 2, 3, 7

[12] RUIWEN CHEN, VALENTINE KABANETS, ANTONINA KOLOKOLOVA, RONEN SHALTIEL, AND DAVID ZUCKERMAN: Mining circuit lower bound proofs for meta-algorithms. *Comput. Complexity*, 24(2):333–392, 2015. Preliminary version in CCC'14. [doi:10.1007/s00037-015-0100-0] 5, 6, 7, 16

[13] DAVID A. COX, JOHN LITTLE, AND DONAL O'SHEA: *Ideals, Varieties and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 1992. [doi:10.1007/978-3-319-16721-3] 20

[14] MERRICK L. FURST, JAMES B. SAXE, AND MICHAEL SIPSER: Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. Preliminary version in FOCS'81. [doi:10.1007/BF01744431] 2

[15] FREDERIC GREEN: A complex-number Fourier technique for lower bounds on the Mod-m degree. *Comput. Complexity*, 9(1):16–38, 2000. [doi:10.1007/PL00001599] 3, 5

[16] VENKATESAN GURUSWAMI, ATRI RUDRA, AND MADHU SUDAN: *Essential Coding Theory*. 2013-17. Available at author's webpage. 18

[17] PRAHLADH HARSHA AND SRIKANTH SRINIVASAN: On polynomial approximations to AC^0. In *Proc. 19th Internat. Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX'16)*, pp. 32:1–32:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.APPROX-RANDOM.2016.32, arXiv:1604.08121] 5

[18] JOHAN HÅSTAD: *Computational Limitations of Small-depth Circuits*. MIT Press, 1987. 2

[19] SWASTIK KOPPARTY AND SRIKANTH SRINIVASAN: Certifying polynomials for $AC^0(\mathit{parity})$ circuits, with applications. In *Proc. 32nd IARCS Ann. Conf. on Foundations of Software Technology and Theoret. Computer Sci. (FSTTCS'12)*, pp. 36–47. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012. [doi:10.4230/LIPIcs.FSTTCS.2012.36] 1, 4, 5

[20] FRANCIS S. MACAULAY: Some properties of enumeration in the theory of modular systems. *Proc. London Math. Soc.*, 26(1):531–555, 1927. [doi:10.1112/plms/s2-26.1.531] 20

[21] ZIPEI NIE AND ANTHONY Y. WANG: Hilbert functions and the finite degree Zariski closure in finite field combinatorial geometry. *J. Combin. Theory Ser. A*, 134:196 – 220, 2015. [doi:10.1016/j.jcta.2015.03.011, arXiv:1402.3018] 5, 16, 17, 19, 20

[22] IGOR CARBONI OLIVEIRA AND RAHUL SANTHANAM: Majority is incompressible by $AC^0[p]$ circuits. In *Proc. 30th Computational Complexity Conf. (CCC'15)*, pp. 124–157. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.CCC.2015.124] 5, 13, 17, 19

[23] PRABHAKAR RAGDE AND AVI WIGDERSON: Linear-size constant-depth polylog-threshold circuits. *Inform. Process. Lett.*, 39(3):143–146, 1991. [doi:10.1016/0020-0190(91)90110-4] 2

[24] ALEXANDER A. RAZBOROV: Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(4):598–607, 1987. 2, 4, 6, 11, 12

[25] ALEXANDER A. RAZBOROV AND STEVEN RUDICH: Natural proofs. *J. Comput. System Sci.*, 55(1):24–35, 1997. Preliminary version in STOC'94. [doi:10.1006/jcss.1997.1494] 18

[26] BENJAMIN ROSSMAN: On the constant-depth complexity of k-clique. In *Proc. 40th STOC*, pp. 721–730. ACM Press, 2008. [doi:10.1145/1374376.1374480] 3, 19

[27] ROMAN SMOLENSKY: Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. 19th STOC*, pp. 77–82. ACM Press, 1987. [doi:10.1145/28395.28404] 2

[28] ROMAN SMOLENSKY: On representations by low-degree polynomials. In *Proc. 34th FOCS*, pp. 130–138. IEEE Comp. Soc. Press, 1993. [doi:10.1109/SFCS.1993.366874] 4, 11, 20

[29] SRIKANTH SRINIVASAN: A compression algorithm for $AC^0[\oplus]$ circuits using certifying polynomials. *Electron. Colloq. on Comput. Complexity (ECCC)*, 22:142, 2015. LINK. 1, 18

[30] AVISHAY TAL: Tight bounds on the Fourier spectrum of AC^0. In *Proc. 32nd Computational Complexity Conf. (CCC'17)*, pp. 15:1–15:31. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.CCC.2017.15] 5

[31] EMANUELE VIOLA: On approximate majority and probabilistic time. *Comput. Complexity*, 18(3):337–375, 2009. Preliminary version in CCC'07. [doi:10.1007/s00037-009-0267-3] 2

[32] INGO WEGENER: *The Complexity of Boolean Functions*. John Wiley & Sons, Inc., 1987. 16

[33] RYAN WILLIAMS: New algorithms and lower bounds for circuits with linear threshold gates. In *Proc. 46th STOC*, pp. 194–202. ACM Press, 2014. [doi:10.1145/2591796.2591858, arXiv:1401.2444] 5

[34] ANDREW CHI-CHIH YAO: Separating the polynomial-time hierarchy by oracles. In *Proc. 26th FOCS*, pp. 1–10. IEEE Comp. Soc. Press, 1985. [doi:10.1109/SFCS.1985.49] 2

AUTHORS

Swastik Kopparty
Associate professor
Department of Mathematics
& Department of Computer Science
Rutgers University
Piscataway, NJ, USA
swastik.kopparty@gmail.com
http://www.math.rutgers.edu/~sk1233

Srikanth Srinivasan
Assistant professor
Department of Mathematics
IIT Bombay, Mumbai, India
srikanth@math.iitb.ac.in
http://www.math.iitb.ac.in/~srikanth

ABOUT THE AUTHORS

SWASTIK KOPPARTY got his Ph. D. in Computer Science from MIT in 2010, and was advised by Madhu Sudan. During 2010-2011, he was a postdoc at the Institute for Advanced Study and he has been at Rutgers University since then. Long before any of this, he got hooked on mathematics early in life because of his mathematician father (and eventual coauthor), K. P. S. Bhaskara Rao. Swastik's research interests are in complexity theory, error-correcting codes, finite fields, randomness and pseudorandomness.

SRIKANTH SRINIVASAN got his undergraduate degree from the Indian Institute of Technology Madras, where his interest in the theory side of CS was piqued under the tutelage of N. S. Narayanswamy. Subsequently, he obtained his Ph. D. from The Institute of Mathematical Sciences in 2011; his advisor was V. Arvind. His research interests span all of TCS (in theory), but in practice are limited to circuit complexity, derandomization, and related areas of mathematics. He enjoys running and pretending to play badminton.