Succinct Hitting Sets and Barriers to Proving Algebraic Circuits Lower Bounds*

Michael A. Forbes†
Simons Institute for the Theory of Computing at the University of California, Berkeley, USA
miforbes@csail.mit.edu

Amir Shpilka‡
Department of Computer Science, Tel Aviv University
Tel Aviv, Israel
shpilka@post.tau.ac.il

Ben Lee Volk§
Department of Computer Science, Tel Aviv University
Tel Aviv, Israel
benleevolk@gmail.com

ABSTRACT

We formalize a framework of *algebraically natural* lower bounds for algebraic circuits. Just as with the natural proofs notion of Razborov and Rudich for boolean circuit lower bounds, our notion of algebraically natural lower bounds captures nearly all lower bound techniques known. However, unlike the boolean setting, there has been no concrete evidence demonstrating that this is a *barrier* to obtaining super-polynomial lower bounds for general algebraic circuits, as there is little understanding whether algebraic circuits are expressive enough to support "cryptography" secure against algebraic circuits.

Following a similar result of Williams in the boolean setting, we show that the existence of an algebraic natural proofs barrier is *equivalent* to the existence of *succinct* derandomization of the polynomial identity testing problem. That is, whether the coefficient vectors of polylog(N)-degree polylog(N)-size circuits is a hitting set for the class of poly(N)-degree poly(N)-size circuits. Further, we give an explicit universal construction showing that *if* such a succinct hitting set exists, then our universal construction suffices.

Further, we assess the existing literature constructing hitting sets for restricted classes of algebraic circuits and observe that *none* of them are succinct as given. Yet, we show how to modify some of these constructions to obtain succinct hitting sets. This constitutes the first evidence supporting the existence of an algebraic natural proofs barrier.

Our framework is similar to the Geometric Complexity Theory (GCT) program of Mulmuley and Sohoni, except that here we emphasize constructiveness of the proofs while the GCT program emphasizes symmetry. Nevertheless, our succinct hitting sets have relevance to the GCT program as they imply lower bounds for the complexity of the defining equations of polynomials computed by small circuits.

CCS CONCEPTS

• **Theory of computation → Algebraic complexity theory**; *Circuit complexity*;

KEYWORDS

Algebraic Circuit Complexity, Polynomial Identity Testing, Barriers, Succinct Hitting Sets

ACM Reference format:
Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. 2017. Succinct Hitting Sets and Barriers to Proving Algebraic Circuits Lower Bounds. In *Proceedings of 49th Annual ACM SIGACT Symposium on the Theory of Computing, Montreal, Canada, June 2017 (STOC'17),* 12 pages.
DOI: 10.1145/3055399.3055496

1 INTRODUCTION

Computational complexity theory studies the limits of efficient computation, and a particular goal is to quantify the power of different computational resources such as time, space, non-determinism, and randomness. Such questions can be instantiated as asking to prove equalities or separations between complexity classes, such as resolving P versus NP. Indeed, there have been various successes: the (deterministic) time-hierarchy theorem showing that $\mathsf{P} \neq \mathsf{EXP}$ ([34]), circuit lower bounds showing that $\mathsf{AC}^0 \neq \mathsf{P}$ ([7, 25, 35, 74]), and interactive proofs showing $\mathsf{IP} = \mathsf{PSPACE}$ ([46, 64]). However, for each of these seminal works we have now established *barriers* for why their underlying techniques *cannot* resolve questions such as P versus NP. Respectively, the above results are covered by the barriers of relativization of Baker, Gill and Solovay [10], natural proofs of Razborov and Rudich [59], and algebraization of Aaronson and Wigderson [3]. In this work we revisit the natural proofs barrier of Razborov and Rudich [59] and seek to understand how it extends to a barrier to algebraic circuit lower bounds. While previous works have considered versions of an algebraic natural proofs barrier, we give the *first* evidence of such a barrier against restricted algebraic reasoning.

Natural Proofs: The setting of Razborov and Rudich [59] is that of *non-uniform* complexity, where instead of considering a Turing machine solving a problem on all input sizes, one considers a model such as boolean circuits where the computational device can change with the size of the input. While circuits are at least as powerful

*A full version of this paper is available at [23].

†This work was performed when the author was at Stanford University, while supported by the NSF, including NSF CCF-1617580, and the DARPA Safeware program.
‡The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575 and from the Israel Science Foundation (grant number 552/16).
§The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575 and from the Israel Science Foundation (grant number 552/16).

STOC'17, Montreal, Canada

DOI: 10.1145/3055399.3055496

as Turing machines, and can even (trivially) compute undecidable languages, their ability to solve computational problems of interest can seem closer to uniform computation. As such, obtaining lower bounds for boolean circuits was seen as a viable method to indirectly tackle Turing machine lower bounds, with the benefit of being able to appeal to more combinatorial methods and thus bypassing the relativization barrier of Baker, Gill and Solovay [10].

There have been many important lower bounds obtained for restricted classes of circuits: constant-depth circuits ([7, 25, 35, 74]), constant-depth circuits with prime modular gates ([58, 67]), as well as lower bounds for monotone circuits ([9, 57, 69]). Razborov and Rudich [59] observed that many of these lower bounds prove *more* than just a lower bound for a single explicit function. Indeed, they observed that such lower bounds often distinguish functions computable by small circuits from *random* functions, and in fact they do so *efficiently*. Specifically, a *natural property* P is a subset of boolean functions $P \subseteq \cup_{n\geq 1}\{f : \{0,1\}^n \to \{0,1\}\}$ with the following properties, where we denote $N := 2^n$ to be the input size to the property.[1]

(1) *Usefulness:* If $f : \{0,1\}^n \to \{0,1\}$ is computable by poly(n)-size circuits then f has property P.
(2) *Largeness:* Random functions $f : \{0,1\}^n \to \{0,1\}$ do not have the property P with noticeable probability, that is, with probability at least $1/\operatorname{poly}(N) = 2^{-O(n)}$.
(3) *Constructivity:* Given a truth-table of a boolean function $f : \{0,1\}^n \to \{0,1\}$, of size $N = 2^n$, deciding whether f has the property P can be checked in $\operatorname{poly}(N) = 2^{O(n)}$ time.

To obtain a circuit lower bound, a priori one only needs to obtain a (non-trivial) property P that is useful in the above sense. However, Razborov and Rudich [59] showed that (possibly after a small modification) most circuit lower bounds (such as those for constant-depth circuits ([7, 25, 35, 58, 67, 74])) yield large and constructive properties, and called such lower bounds *natural proofs*.

Further, Razborov and Rudich [59] argued that standard cryptographic assumptions imply that natural proofs *cannot* yield super-polynomial lower bounds against any restricted class of circuits that is sufficiently rich to implement cryptography. That is, a *pseudo-random function* is an efficiently computable function $f : \{0,1\}^n \times \{0,1\}^\lambda \to \{0,1\}$ such that when sampling the key $\mathrm{k} \in \{0,1\}^\lambda$ at random the resulting distribution of functions $f(\cdot, \mathrm{k})$ is computationally indistinguishable from a truly random function $\mathrm{f} : \{0,1\}^n \to \{0,1\}$. The existence of pseudorandom functions follows from the existence of *one-way functions* ([27, 36]) which is essentially the weakest interesting cryptographic assumption. There are even candidate constructions of pseudorandom functions computable by polynomial-size constant-depth threshold circuits (TC^0) as given by Naor and Reingold [49], whose security rests on the intractability of discrete-log and factoring-type assumptions (see also Krause and Lucks [44]). As such, it is widely-believed that there are pseudorandom functions, even ones computationally indistinguishable from random except to adversaries running in $\exp(\lambda^{\Omega(1)})$-time.

[1]The Razborov and Rudich [59] definition of a natural property actually applies to the complement of the property P we use here. This is a trivial difference for boolean complexity, but is important for algebraic complexity as there natural properties are one-sided.

In contrast, Razborov and Rudich [59] showed that a natural proof useful against poly(n)-size circuits can distinguish a pseudorandom function from a truly random function in $\operatorname{poly}(2^n)$-time, which would contradict the believed $\exp(\lambda^{\Omega(1)})$-indistinguishability when taking λ to be a large enough polynomial in n. That is, suppose P is a natural property. Then for a pseudorandom function $f(\cdot,\cdot)$ and each value $k \in \{0,1\}^\lambda$ of the key, the resulting function $f(\cdot, k) : \{0,1\}^n \to \{0,1\}$ has a poly(n)-size circuit, and has property P (by usefulness). In contrast, random functions will not have property P with noticeable probability (by largeness). As the property is constructive, this gives a $\operatorname{poly}(2^n)$-time algorithm distinguishing $f(\cdot, \mathrm{k})$ from a random function, as desired.

Algebraic Natural Proofs: Algebraic circuits are the most natural model for computing polynomials by using addition and multiplication. While more restricted than general (boolean) computation, proving lower bounds for algebraic circuits has proved challenging. Yet, we do not have formal barrier results for understanding the difficulty of such lower bounds. While such lower bounds are not a priori subject to the natural proofs barrier due to the formal differences in the computational model, the relevance of the ideas of natural proofs to algebraic circuits has been repeatedly asked. Aaronson-Drucker [2] as well as Grochow [28] noticed that many of the prominent algebraic circuit lower bounds (such as [50–52, 56]) are *algebraically natural*, in that they obey an algebraic form of usefulness, largeness, and constructivity.

While this would seemingly imply a Razborov and Rudich [59]-type barrier for existing techniques, there is a key piece missing: we have very little evidence for the existence of *algebraic* pseudorandom functions. That is, the pseudorandom functions used by Razborov and Rudich [59] are *boolean* functions, and naive attempts to algebrize them seemingly do not yield pseudorandom polynomials. Indeed, as algebraic circuits are a computational model weaker than general computation, it is conceivable that they are too weak to implement cryptography, so that natural proofs barrier would *not* apply. In contrast, it is also conceivable that algebraic circuits are so weak that they can compute "enough" cryptography to be secure against algebraic circuits, so that a natural proofs barrier *would* apply.

Our Work: In this work we formalize the study of pseudorandom polynomials by exhibiting the *first* constructions provably secure against restricted classes of algebraic circuits. In particular, we follow Williams [73] in treating the existence of a natural proofs barrier as the problem of *succinct* derandomization: replacing randomness with pseudorandomness that further has a *succinct* description. We revisit existing derandomization of restricted classes of algebraic circuits and show (via non-trivial modification) that they can be made succinct in many cases.

Recently, and independently of our work, Grochow, Kumar, Saks, and Saraf [29] observed a similar connection between a natural proofs barrier for algebraic circuits and succinct derandomization. Their work also presents connections with Geometric Complexity Theory (which we discuss below in Section 1.7) and algebraic proof complexity. However, unlike our work they do not present any constructions of succinct derandomization.

1.1 Algebraic Complexity

We now discuss the algebraic setting for which we wish to present the natural proofs barrier. Algebraic complexity theory studies the complexity of syntactic computation of polynomials using algebraic operations. The most natural model of computation is that of an *algebraic circuit*, which is a directed acyclic graph whose leaves are labeled by either variables $x_1, \ldots, x_n$ or elements from the field $\mathbb{F}$, and whose internal nodes are labeled by the algebraic operations of addition (+) or multiplication (×). Each node in the circuit computes a polynomial in the natural way, and the circuit has one or more *output nodes*, which are nodes of out-degree zero. The *size* of the circuit is defined to be the number of wires, and the *depth* is defined to be the length of a longest path from an input node to an output node. As usual, a circuit whose underlying graph is a tree is called a *formula*. One can associate various complexity classes with algebraic circuits, and the most important one for us is VP, which the classes of n-variate polynomials with poly(n)-degree computable by poly(n)-size algebraic circuits. There is also VNP, which we will informally define as the class of "explicit" polynomials.

A central open problem in algebraic complexity theory is proving a super-polynomial lower bound for the algebraic circuit size of any explicit polynomial, that is, proving VP ≠ VNP. Substantial attention has been given to this problem, using various techniques that leverage non-trivial algebraic tools to study the syntactic nature of these circuits. Indeed, our knowledge of algebraic lower bounds seem to surpass that of boolean circuits, as we have super-linear lower bounds for general circuits ([11, 68]) — a goal as yet unachieved in the boolean setting. Similarly, there are a wide array of super-polynomial or even exponential lower bounds known for various weaker models of computation such as non-commutative formulas ([50]), multilinear formulas ([53, 55]), and homogeneous depth-3 and depth-4 circuits ([24, 32, 40, 41, 45, 51]). We refer the reader to Saptharishi [60] for a continuously-updating comprehensive compendium of these lower bounds.

However, this landscape might feel reminiscent of the boolean setting, in that there are various restricted models where lower bounds techniques are known, and yet lower bounds for general circuits or formulas remain relatively poorly understood. Yet, there has been some significant recent cause for optimism for obtaining *general* circuit lower bounds, as various depth-reduction results ([6, 8, 15, 32, 43, 70, 72]) have shown that n-variable degree-d polynomials computable by size-s algebraic circuits have $s^{O(\sqrt{d})}$-size depth-3 or homogeneous depth-4 formulas. Further, recent methods ([24, 31, 39–41, 45]) have proven $(nd)^{\Omega(\sqrt{d})}$ lower bounds computing explicit polynomials by homogeneous depth-4 formulas. If one could simply push these methods to obtain an $(nd)^{\omega(\sqrt{d})}$ lower bound then this would obtain super-polynomial lower bounds for general circuits! Unfortunately, all of the lower bounds methods known seem to apply not just to candidate hard polynomials, but also to certain *easy* polynomials, demonstrating that these techniques cannot yield a $(nd)^{\omega(\sqrt{d})}$ lower bound as this would contradict the depth-reduction theorems.

Given this state of affairs, it is unclear whether to be optimistic or pessimistic regarding future prospects for obtaining superpolynomial lower bounds for general algebraic circuits. To resolve this uncertainty it is clearly important to formalize the barriers constraining our lower bound techniques. Indeed, as mentioned above all known lower-bound methods apply not just to hard polynomials but also to easy polynomials — is this intrinsic to current methods? This is essentially the question of whether there is an algebraic natural proofs barrier, as we now describe.

1.2 Algebraic Natural Proofs

We now define the notion of an *algebraically natural proof* used in this paper. Intuitively, we want to know whether lower bounds methods can distinguish between polynomials of low-complexity and polynomials of high-complexity, so that they are *useful* in the sense of Razborov and Rudich [59]. In particular, we want to know if such distinguishers can be *efficient*, so that they are also *constructive*. Several works, such as Aaronson and Drucker [2], Grochow [28] (see also Shpilka and Yehudayoff [66, Section 3.9], and Aaronson [1, Section 6.5.3]) have noticed that most all of the lower bounds methods in algebraic complexity theory are themselves algebraic in a certain sense which we now describe.

The simplest example is to consider matrix rank, where the complexity of an $n \times n$ matrix M is exactly captured by its determinant, which is a polynomial. That is, if M is of rank $< n$ then $\det M = 0$, and if rank $= n$ then $\det M \neq 0$. The key feature here is that $\det M$ is a polynomial in the *coefficients of the underlying algebraic object*, which in this case is the matrix M. Most of the central lower bounds techniques, such as partial derivatives ([51]), evaluation/coefficient dimension ([22, 50, 52, 56]), or shifted partial derivatives ([31, 39]) are generalizations of this idea, specifically leveraging notions of linear algebra and rank. Abstractly, these methods take an n-variate polynomial f, inspect its coefficients, and then form an exponentially-large (in n) matrix M_f whose entries are polynomials in the coefficients of f. One then shows that if f is simple then $\operatorname{rank} M_f < r$, while for an explicit polynomial f_0 one can show that $\operatorname{rank} M_{f_0} \geq r$. In particular, by basic linear algebra this shows that there is some $r \times r$ submatrix M'_{f_0} of M_{f_0} such that $\det M'_{f_0} \neq 0$, and yet $\det M'_f = 0$ for simple f, where M'_f denotes the restriction of M_f to the same set of rows and columns. This proves that f_0 is a hard polynomial.

We now observe that the above outline gives a natural property $P := \{f : \det M'_f = 0\}$ in the sense of Razborov and Rudich [59].

(1) *Usefulness:* For low-complexity f we have that $f \in P$ as argued above. Further, P is a non-trivial property as $f_0 \notin P$.
(2) *Constructivity:* For a given f, deciding whether "$f \in P$?" is tantamount to computing $\det M'_f$. Even though M'_f might be exponentially-large, it is often polynomially-large in the *size of* f (which is exponential in the number n of variables in f). As typically M'_f is a simple matrix in terms of f, computing $\det M'_f$ is essentially the complexity of computing the determinant, which is computable by small algebraic circuits ([12, 47]). Thus, the property P is efficiently decidable in the size of its input.
(3) *Largeness:* The largeness condition is *intrinsic* here, as the property is governed by the vanishing of a non-zero polynomial; $\det M'_f$ is non-zero as a polynomial as in particular

$\det M'_{f_0} \neq 0$. As non-zero polynomials evaluate to non-zero at random points with high probability ([16, 63, 75]), this means that such distinguishers certify that random polynomials are of high-complexity.

Thus, we see that the above meta-method forms a very natural instance of a natural property. As such, one might expect the Razborov and Rudich [59] barrier to then rule out such properties, however their barrier result only holds when the underlying circuit class can compute pseudorandom functions. While it is widely believes that simple boolean circuit classes can compute pseudorandom functions (as discussed above), the ability of algebraic circuits to compute pseudorandom functions is significantly less understood. As such, the Razborov and Rudich [59] barrier's applicability to the algebraic setting is not immediate. However, as the above meta-method obeys algebraic restrictions on the natural properties being considered, this suggests that barrier could follow from a weaker assumption than that of algebraic circuits computing pseudorandom functions.

We now give a formalization of the above meta-method for algebraic circuit lower bounds, which is implicit in prior work and known to experts. To begin, we must first note that in comparing low-complexity to high-complexity polynomials, we must detail the space in which the polynomials reside. There are three spaces of primary interest.

(1) $\mathbb{F}[x_1, \ldots, x_n]^d$: The space of n-variate polynomials of total degree at most d. There are $N_{n,d} := \binom{n+d}{d}$ many monomials $\mathbf{x}^{\mathbf{a}} := x_1^{a_1} \cdots x_n^{a_n}$ in this space.
(2) $\mathbb{F}[x_1, \ldots, x_n]^d_{\text{hom}}$: The space of homogeneous n-variate polynomials of total degree exactly d. There are $N^{\text{hom}}_{n,d} := \binom{n+d-1}{d}$ many monomials $\mathbf{x}^{\mathbf{a}}$ in this space.
(3) $\mathbb{F}[x_1, \ldots, x_n]^d_{\text{ideg}}$: The space of n-variate polynomials of individual degree at most d. There are $N^{\text{ideg}}_{n,d} := (d+1)^n$ many monomials $\mathbf{x}^{\mathbf{a}}$ in this space.

We now present our definition, with enough generality to handle the above spaces of polynomials simultaneously. That is, for a fixed set of monomials $\mathcal{M}$ (such as all monomials of degree at most d) we consider the space $\text{span}(\mathcal{M})$, which is defined as all linear combinations over monomials in $\mathcal{M}$. We then identify a polynomial $f \in \text{span}(\mathcal{M})$ defined by $f = \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}$ with its list of such coefficients, which is a vector $\mathbf{coeff}_{\mathcal{M}}(f) \in \mathbb{F}^{\mathcal{M}}$ defined $\mathbf{coeff}_{\mathcal{M}}(f) := (c_{\mathbf{a}})_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}}$. We then ask for distinguisher D which take as input these $|\mathcal{M}|$ many coefficients, which can separate low-complexity polynomials from high-complexity polynomials.

Definition 1.1 (Algebraically Natural Proof). Let $\mathcal{M} = \{\mathbf{x}^{\mathbf{a}}\}_{\mathbf{a}} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be a set of monomials, and let the set $\text{span}(\mathcal{M}) := \{\sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F}\}$ be all linear combinations of these monomials. Let $C \subseteq \text{span}(\mathcal{M})$ and $\mathcal{D} \subseteq \mathbb{F}[\{c_{\mathbf{a}}\}_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}}]$ be classes of polynomials, where the latter is in $|\mathcal{M}|$ many variables.

A polynomial $D \in \mathcal{D}$ is an **algebraic $\mathcal{D}$-natural proof against** C, also called a **distinguisher**, if

(1) D is a non-zero polynomial.
(2) For all $f \in C$, D vanishes on the coefficient vector of f, that is, $D(\mathbf{coeff}_{\mathcal{M}}(f)) = 0$.

We will be primarily interested in taking the set of monomials M to correspond to one of the above three sets of polynomials, $\mathbb{F}[\mathbf{x}]^d$, $\mathbb{F}[\mathbf{x}]^d_{\text{hom}}$ and $\mathbb{F}[\mathbf{x}]^d_{\text{ideg}}$, to which we define the relevant coefficient vectors as $\mathbf{coeff}_{n,d}$, $\mathbf{coeff}^{\text{hom}}_{n,d}$ and $\mathbf{coeff}^{\text{ideg}}_{n,d}$. We will use "$\mathbf{coeff}$" if the space of polynomials is clear from the context.

Thus, to revisit the comparison with Razborov and Rudich [59], condition (2) says that the distinguisher D is *useful* against the class C. Condition (1) indicates that the property is non-trivial, and in particular is *large*, as a non-zero polynomial will evaluate to non-zero at a random point with high probability ([16, 63, 75]). Finally, the fact that distinguisher D comes from the restricted class $\mathcal{D}$ is the *constructivity* requirement, and the main question is how simple the distinguisher D can be.

We argued above that most of the main lower bound techniques fall into the above algebraic natural proof paradigm where the distinguisher has polynomial-size algebraic circuits, so that the proof is VP-natural. This motivates the following question about algebraic VP-natural proofs against VP.

Question 1.2. *For the space of n-variate total degree d polynomials $\mathbb{F}[x_1, \ldots, x_n]^d$, is there an algebraic* $\text{poly}(N_{n,d})$*-size natural proof for lower bounds against* $\text{poly}(n, d)$*-size circuits?*

While one could make a detailed study of existing lower bounds to prove the intuitive fact that VP-natural properties suffice for them, our attention will be to studying the *limits* of this framework. That said, it is worth mentioning that there are known techniques for algebraic circuit lower bounds that fall outside this framework.

First, the shifted partial derivative technique of Gupta, Kamath, Kayal and Saptharishi [31, 39] is not currently known to be VP-natural. That is, while it does fall into the above rank-based meta-method (and thus the algebraic natural proof paradigm), the matrices involved are actually *quasi*-polynomially large in their input, so the method is only quasiVP-natural. However, as the shifted partial technique proves exponential lower bounds the required quasiVP-naturalness still seems rather modest.

In contrast, there are actually methods which *completely* fall of the algebraic framework (constructive or not). That is, as discussed below in Section 1.7, this algebraic distinguisher framework is limited to proving *border* complexity lower bounds, where border complexity is always upper bounded by usual complexity notions. For the *tensor rank* model, distinguishers actually prove border rank lower bounds. In contrast, the substitution method ([14, Chapter 6],[13]) can prove tensor rank lower bounds which are *higher* than known border rank upper bounds (for explicit tensors), giving a separation between these two complexities and thus showing the substitution method is not captured by the algebraic natural proof framework. However, all such known separations are by at most a multiplicative constant factor, so the inability of the substitution method to be algebraically natural does not currently seem to be a serious deficiency in the framework developed here.

1.3 Pseudorandom Polynomials

Having given our formal definition of algebraic natural proofs, we now explain our notion of the algebraic natural proof *barrier*. In

particular, as algebraically natural proofs concern the zeros of (non-zero) polynomials computable by small circuits, this naturally leads us to the *polynomial identity testing (PIT)* problem.

Polynomial identity testing is the following algorithmic problem: given an algebraic circuit D computing an N-variate polynomial, decide whether D computes the identically zero polynomial. The problem admits a simple efficient randomized algorithm by the Schwartz-Zippel-DeMillo-Lipton Lemma [16, 63, 75]. That is, evaluations of a low-degree non-zero polynomial at random points taken from a large enough field will be non-zero with high probability. Thus, to check non-zeroness it is enough to evaluate D on a random input $\boldsymbol{\alpha}$ and observe whether $D(\boldsymbol{\alpha}) = 0$, which is clearly efficient. However, the best known deterministic algorithms run in exponential time. Designing an efficient deterministic algorithm for PIT is another major open problem in algebraic complexity, with intricate and bidirectional connections to proving algebraic and boolean circuit lower bounds [4, 17, 37, 38].

The two flavors in which the problem appears are the *white-box* model, in which the algorithm is allowed to inspect the structure of the circuit, and the *black-box* model, in which the algorithm is only allowed to access evaluations of the circuit on inputs of its choice, such as the randomized algorithm described above. It can be easily seen that efficient deterministic black-box algorithms are equivalent to constructing small *hitting sets*: a hitting set for a class $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_N]$ of circuits is a set $\mathcal{H} \subseteq \mathbb{F}^N$ such that for any non-zero circuit $D \in \mathcal{D}$, there exists $\boldsymbol{\alpha} \in \mathcal{H}$ such that $D(\boldsymbol{\alpha}) \neq 0$. While small hitting sets *exist* for VP, little progress has been made for explicitly constructing any non-trivial hitting sets for general algebraic circuits (or even solving PIT in the white-box model). In contrast, there has been substantial work developing efficient deterministic white- and black-box PIT algorithms for non-trivial restricted classes of algebraic computation, see for example the surveys of Saxena [61, 62] and Shpilka-Yehudayoff [66].

We now define our notion of pseudorandom polynomials by connecting the algebraic natural proof framework with hitting sets. Consider a class C of polynomials, say in the space of polynomials of bounded total degree $\mathbb{F}[x_1, \ldots, x_n]^d$. If D is an algebraic natural proof against C then we have:

(1) D is a non-zero polynomial.
(2) D vanishes on the set $\mathcal{H} := \{\mathbf{coeff}_{n,d}(f) : f \in C\}$ of coefficient vectors of polynomials in C.

Put together, these conditions are equivalent to saying that that $\mathcal{H}$ is *not* a hitting set for D. Thus, we see that there are algebraically natural proofs *if and only if* coefficient-vectors of simple polynomials are *not* hitting sets. Thus, we see that the existence of an algebraic natural proofs barrier can be rephrased as whether PIT can be derandomized using *succinct* pseudorandomness. A completely analogous statement was proven by Williams [73] in the boolean setting, where the existence of the Razborov and Rudich [59] natural proofs barrier was shown equivalent to succinct derandomization of ZPE, those problems solvable in zero-error $2^{O(n)}$-time. However, that equivalence there is slightly more involved, while it is immediate here.

We now give the formal definition mirroring the above discussion, in the same generality of Definition 1.1.

Definition 1.3 (Succinct Hitting Set). Let $\mathcal{M} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be a set of monomials $\mathcal{M} = \{\mathbf{x}^{\mathbf{a}}\}_{\mathbf{a}}$, and let the set $\mathrm{span}(\mathcal{M}) := \{\sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F}\}$ be all linear combinations of these monomials. Let $C \subseteq \mathrm{span}(\mathcal{M})$ and $\mathcal{D} \subseteq \mathbb{F}[\{c_{\mathbf{a}}\}_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}}]$ be classes of polynomials, where the latter is in $|\mathcal{M}|$ many variables.

C is a **C-succinct hitting set for $\mathcal{D}$** if $\mathcal{H} := \{\mathbf{coeff}_{\mathcal{M}}(f) : f \in C\}$ is a hitting set for $\mathcal{D}$. That is, $D \in \mathcal{D}$ is non-zero iff $D|_{\mathcal{H}}$ is non-zero, that is, there is some $f \in C$ such that $D(\mathbf{coeff}_{\mathcal{M}}(f)) \neq 0$.

To make our statements more concise, we often abbreviate the name of the class C in a way which is understood from the context. For example, the modifier "s-succinct", with s being an integer, will refer to a C-hitting set with C being the class of circuits of size at most s. Similarly, s-$\Sigma\Pi\Sigma$-succinct will refer to C being the class of depth-3 circuits of size at most s, and so on.

The above argument showing the tension between algebraic natural proofs and pseudorandom polynomials can be summarized in the following theorem, which follows immediately from the definitions.

Theorem 1.4. *Let $\mathcal{M} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be a set of monomials $\mathcal{M} = \{\mathbf{x}^{\mathbf{a}}\}_{\mathbf{a}}$, and let the set $\mathrm{span}(\mathcal{M}) := \{\sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F}\}$ be all linear combinations of these monomials. Let $C \subseteq \mathrm{span}(\mathcal{M})$ and $\mathcal{D} \subseteq \mathbb{F}[\{c_{\mathbf{a}}\}_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}}]$ be classes of polynomials, where the latter is in $|\mathcal{M}|$ many variables.*

Then there is an algebraic $\mathcal{D}$-natural proof against C iff C is not a C-succinct hitting set for $\mathcal{D}$.

Instantiating this claim with $\mathcal{M}$ being the space of degree-d monomials, we get the following quantitative version of the above.

Corollary 1.5. *Let $C \subseteq \mathbb{F}[x_1, \ldots, x_n]^d$ be the class of $\mathrm{poly}(n, d)$-size circuits of total degree at most d. Then there is an algebraic $\mathrm{poly}(N_{n,d})$-natural proof against C iff C is not a $\mathrm{poly}(n, d)$-succinct hitting set for $\mathrm{poly}(N_{n,d})$-size circuits in $N_{n,d}$ variables.*

In the common regime when $d = \mathrm{poly}(n)$, we have that $\mathrm{poly}(n) = \mathrm{polylog}(N_{n,d})$. That is, this existence of an algebraic natural proofs barrier is equivalent to saying that coefficient vectors of circuits of polylogarithmic size (in polylogarithmic many variables) form a hitting set of polynomial-size.

With this equivalence in hand, we can now phrase the question of an algebraic natural proofs barrier.

Question 1.6 (Algebraic Natural Proofs Barrier). *Is there a $\mathrm{polylog}(N)$-succinct hitting set for circuits of $\mathrm{poly}(N)$-size?*

Again, we note that Question 1.6 was also raised by Grochow, Kumar, Saks, and Saraf [29], who presented a definition similar to Definition 1.3 and also observed the implication in Theorem 1.4.

While the above equivalence already suffices for studying the barrier, the notion of a hitting set is sometimes fragile. A more robust way to obtain hitting sets for a class $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_N]$ is to obtain a *generator*, which is a polynomial map $\boldsymbol{\mathcal{G}} : \mathbb{F}^{\ell} \to \mathbb{F}^N$ such that $D \in \mathcal{D}$ is a non-zero iff $D \circ \boldsymbol{\mathcal{G}} \not\equiv 0$, that is, the composition $D(\boldsymbol{\mathcal{G}}(\mathbf{y})) \neq 0$ is non-zero as a polynomial in $\mathbf{y}$. Here one measures the quality of the generator by asking to minimize the seed-length ℓ. By polynomial interpolation, it follows that constructing small hitting sets is equivalent to constructing generators with ℓ small, see for example Shpilka-Yehudayoff [66].

However, in our setting we want *succinct* generators so that the polynomial-map $\mathcal{G}$ is a coefficient vector of a polynomial $G(\mathbf{x}, \mathbf{y})$ computable by a small algebraic circuit. In particular, converting a succinct hitting set $\mathcal{H}$ to a generator using the standard interpolation methods would give a generator which has circuit size $\text{poly}(|\mathcal{H}|)$. However, as we are trying to hit polynomials on N variables, this would yield a $\text{poly}(N)$-size generator whereas we would want a generator of complexity $\text{polylog}(N)$. As such, we now define succinct generators and give a tighter relationship with succinct hitting sets.

Definition 1.7. Let $\mathcal{M} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be a set of monomials $\mathcal{M} = \{\mathbf{x}^{\mathbf{a}}\}_{\mathbf{a}}$, and let the set $\text{span}(\mathcal{M}) := \{\sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F}\}$ be all linear combinations of these monomials. Let $C \subseteq \text{span}(\mathcal{M})$ and $\mathcal{D} \subseteq \mathbb{F}[\{c_{\mathbf{a}}\}_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}}]$ be classes of polynomials, where the latter is in $|\mathcal{M}|$ many variables. Further, let $C' \subseteq \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_\ell]$ be another class of polynomials.

We say that a polynomial map $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^{\mathcal{M}}$ is a **C-succinct generator for $\mathcal{D}$ computable in C'** if

(1) The polynomial $G(\mathbf{x}, \mathbf{y}) := \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} \mathcal{G}_{\mathbf{x}^{\mathbf{a}}}(\mathbf{y}) \cdot \mathbf{x}^{\mathbf{a}}$ is a polynomial from C', where $\mathcal{G}_{\mathbf{x}^{\mathbf{a}}}(\mathbf{y})$ is the polynomial computed by the $\mathbf{x}^{\mathbf{a}}$-coordinate of $\mathcal{G}$.
(2) For every value $\boldsymbol{\alpha} \in \mathbb{F}^\ell$, the polynomial $G(\mathbf{x}, \boldsymbol{\alpha}) \in C$.
(3) $\mathcal{G}$ is a generator for for $\mathcal{D}$. That is, $D \in \mathcal{D}$ is a non-zero polynomial in $\mathbb{F}[\mathbf{c}]$ iff $D \circ \mathcal{G} \not\equiv 0$ in $\mathbb{F}[\mathbf{y}]$, meaning that $D(\textbf{coeff}_{\mathcal{M}}(G(\mathbf{x}, \mathbf{y}))) \neq 0$ as a polynomial in $\mathbb{F}[\mathbf{y}]$, where we think of $G(\mathbf{x}, \mathbf{y})$ as a polynomial in the ring $(\mathbb{F}[\mathbf{y}])[\mathbf{x}]$ and take these coefficients with respect to the $\mathbf{x}$ variables, so that $\textbf{coeff}_{\mathcal{M}}(G(\mathbf{x}, \mathbf{y})) \in \mathbb{F}[\mathbf{y}]^{\mathcal{M}}$.

Conditions (2) and (3) are equivalent, over large enough fields, to the property that the output of the generator $\mathcal{G}(\mathbf{x}, \mathbb{F}^\ell) = \{G(\mathbf{x}, \boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{F}^\ell\}$ is a C-succinct hitting set for $\mathcal{D}$. However, the generator result is a priori stronger as it says that the hitting set can be succinctly indexed by a polynomial in C'.

Also, note that the C' computability of the generator implies C'-succinctness, that is, that its image $\{G(\mathbf{x}, \boldsymbol{\alpha}) : \boldsymbol{\alpha} \in \mathbb{F}^\ell\}$ are all circuits which are C'-circuits, at least assuming that C' is a class of polynomials which is closed under substitution. However, sometimes the actual succinctness C can be more stringent than C' for restricted classes of computation. Since the implication regarding barriers to lower bounds only concerns the class C, we often omit mentioning C' and only talk about C-succinct generators for $\mathcal{D}$.

We now give our first result, which uses the construction of a universal circuit to show that there is an explicit universal construction of a succinct generator, that is, this circuit is a succinct generator if there are *any* succinct hitting sets. Further, this shows that *any* succinct hitting set (even infinite) implies a quasipolynomial deterministic black-box PIT algorithm. To make this theorem clear, let VP_m denote the class of small low-degree circuits in m variables.

Theorem 1.8 (Informal summary of Section 3). *There is an explicit* $\text{polylog}(N)$*-size circuit which is a* $\text{VP}_{\text{polylog}(N)}$*-succinct generator for* VP_N *iff there is any* $\text{VP}_{\text{polylog}(N)}$*-succinct hitting set for* VP_N*. Further, the existence of any* $\text{VP}_{\text{polylog}(N)}$*-succinct hitting set for* VP_N *implies an explicit* $\text{poly}(N)^{\text{polylog}(N)}$*-size hitting set for* VP_N*.*

1.4 Evidence for Pseudorandom Polynomials and Our Results

Having now given our formalization of algebraic natural proofs and the corresponding barrier, we now investigate evidence for such barriers. To understand these barriers, it is helpful to remind ourselves of the evidence in the boolean setting.

Boolean Complexity: When speaking of a natural proofs barrier, it is helpful to remember that such barriers are inherently *conditional* (as opposed to relativization ([10]) and algebraization ([3]), which are unconditional). As such, our belief in such barriers rests on the plausibility of these conditional assumptions, for which there exist both cryptographic and complexity-theoretic evidence.

Unfortunately, evidence for an algebraic natural proofs has been much more difficult to obtain. Indeed, the cryptographic evidence in the boolean setting seems less relevant to the algebraic world. Direct attempts to algebrize the underlying cryptographic objects will only yield *functions* that seem pseudorandom, where as we need *polynomials*. While our universal construction (Section 3) gives a universal candidate pseudorandom polynomial, we lack the corresponding web of reductions that reduces the analysis of such candidates to more traditional and well-studied conjectures. In particular, the construction of Goldreich, Goldwasser and Micali [27] that converts a pseudorandom generator to a pseudorandom function seems to have no algebraic analogue ([2]) as this construction applied to polynomials produces polynomials of exponential degree and thus do not live in the desired space of low-degree polynomials $\mathbb{F}[x_1, \ldots, x_n]^d$.

Given the complete lack of algebraic-cryptographic evidence for an algebraic natural proofs barrier, it is then natural to turn to *complexity-theoretic* evidence in the form of succinct derandomization, which constitutes our results.

1.5 Our Results

In this work we present the first unconditional succinct derandomization of various restricted classes of algebraic computation, giving the first evidence at all for an algebraic natural proofs barrier. It is worth noting that in the boolean setting, as discussed above, many derandomization results are *already* succinct. It turns out that, to the best of our knowledge, all existing derandomization for restricted algebraic complexity classes are *not* succinct.

A primary reason for this is that to obtain the best derandomization for polynomials, one typically wants to use univariate generators as this produces more randomness-efficient results (much in the same way that univariate Reed-Solomon codes have better distance than multi-variate Reed-Muller codes). However, univariate polynomials are not VP-succinct essentially by definition as VP looks for multivariate polynomials where the degree is commensurate with the number of variables. Another reason is that while hardness-vs-randomness can produce succinct derandomization in the boolean setting as mentioned above, the known algebraic hardness-vs-randomness paradigm ([38]) is much harder to instantiate for restricted classes of algebraic computation.

However, it seems highly plausible that by redoing existing constructions one can obtain succinct derandomization, and as such we posit the following meta-conjecture.

Meta-Conjecture 1.9. *For any class $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_N]$ for which explicit constructions of subexponential-size hitting sets are currently known, there are subexponential-size hitting-sets which are* $\mathrm{polylog}(N)$*-succinct, where succinctness is measured with respect to one of the spaces of polynomials $\mathbb{F}[x_1, \ldots, x_n]^d$, $\mathbb{F}[x_1, \ldots, x_n]^d_{\mathrm{hom}}$, or $\mathbb{F}[x_1, \ldots, x_n]^d_{\mathrm{ideg}}$.*

In this work we establish this meta-conjecture for many, but not all, known derandomization results for restricted classes of algebraic circuits. We obtain succinctness with respect to computations in the space of multilinear polynomials $\mathbb{F}[x_1, \ldots, x_n]^1_{\mathrm{ideg}}$. In some cases similar results could be obtained with respect to the space of total degree $\mathbb{F}[x_1, \ldots, x_n]^d$, but we omit discussion of these techniques as the $\mathbb{F}[x_1, \ldots, x_n]^1_{\mathrm{ideg}}$ results are cleanest. All of our succinct derandomization results will be via succinct generators, but as the hitting sets have succinctness even beyond the succinctness of the generator we will focus on presenting the succinctness of the hitting sets instead.

We now list our results, but omit the exact definitions of these models which apper in the full version. We begin with succinct derandomization covering many of the hitting-set constructions for constant-depth circuits with various restrictions. These formulas will be fooled by hitting sets which are themselves depth-3 formulas, but of polylogarithmic complexity.

Theorem 1.10. *In the polynomial space of multilinear polynomials $\mathbb{F}[x_1, \ldots, x_n]^1_{\mathrm{ideg}}$, the set of* $\mathrm{poly}(\log s, n)$*-size multilinear $\Sigma\Pi\Sigma$ formulas is a succinct hitting set for $N = 2^n$-variate size-s computations of the form*

- $\Sigma^{O(1)}\Pi\Sigma$ *formulas*
- $\Sigma\Pi\Sigma$ *formulas of transcendence degree* $\leq O(1)$
- *Sparse polynomials*
- $\Sigma\mathrm{m} \wedge \Sigma\Pi^{O(1)}$*-formulas*
- *Commutative roABPs*
- *Depth-$O(1)$ Occur-$O(1)$ formulas*
- *Arbitrary circuits composed with sparse polynomials of transcendence degree $O(1)$*

We now conclude with a weaker result, which is not truly succinct in that the hitting set is of complexity commensurate with the class being fooled. However, this result is for fooling classes of algebraic computation which while restricted, go beyond constant-depth formulas, and as such our result is still non-trivial. This class of computation is known as *read-once oblivious algebraic branching programs (roABPs)*, which can be seen as an algebraic version of RL.

Theorem 1.11 (Section 6). *In the space of multilinear polynomials $\mathbb{F}[x_1, \ldots, x_n]^1_{\mathrm{ideg}}$, the set of width-$w^2$ length-n roABPs is a succinct hitting set for width-w and length-$N = 2^n$ roABPs with a monomial compatible ordering of the variables.*

1.6 Techniques

We now discuss the techniques we use to obtain our succinct hitting sets. The first technique is to carefully choose *which* existing hitting sets constructions to make succinct. In particular, one would naturally want to start with the simplest restricted classes of circuits to fool, which would be sparse polynomials. A well-known hitting-set construction is due to Klivans and Spielman [42], which is often used in hitting-set constructions for more sophisticated algebraic computation. However, as we explain in Section 7, it actually seems difficult to obtain a succinct version of this hitting set (or variants of it).

Instead, we observe that, due to the results of Section 3 mentioned above, we need not focus on the *size* of the hitting sets but rather only on their succinctness. That is, to obtain succinct hitting sets for s-sparse polynomials we need not look at the $\mathrm{poly}(s)$-size hitting sets of Klivans and Spielman [42] but can also consider $\mathrm{poly}(s)^{\mathrm{polylog}(s)}$-size hitting sets which may be more amenable to being made succinct. In particular, there is a generator of Shpilka and Volkovich [65] which can be seen as an algebraic analogue of k-wise independence. It has been shown that this generator fools sparse polynomials with a hitting set of $\mathrm{poly}(s)^{\mathrm{polylog}(s)}$-size, and we we show how to modify this result so the generator is also succinct. Similarly, there is a family of hitting sets which use the *rank condensers* of Gabizon and Raz [26] to produce a pseudorandom linear map that reduces from n variables down to $r \ll n$ variables. We also suitably modify this construction to be succinct. Between these two core constructions, as well as their combination, we are able to make succinct much of the existing hitting set literature.

1.7 Algebraic Natural Proofs and Geometric Complexity Theory

We now comment on the connection between algebraic natural proofs and the Geometric Complexity Theory (GCT) program of Mulmuley and Sohoni [48]. This program posits a very well motivated method for obtaining algebraic circuit lower bounds, drawing inspiration from algebraic geometry and representation theory.

To begin, we briefly discuss some algebraic geometry, so that we now work over an algebraically closed field $\mathbb{F}$. Suppose we have a class of polynomials $C \subseteq \mathbb{F}[x_1, \ldots, x_n]^d$, which we can thus think of as vectors in the space $\mathbb{F}^{N_{n,d}}$. As we did before, we can look at classes of distinguisher polynomials $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_{N_{n,d}}]$ which take as inputs the vector of coefficients of a polynomial in $\mathbb{F}[x_1, \ldots, x_n]^d$. In particular, we wish to look at the class of distinguishers D that vanish on all of C, that is $\mathcal{D} = \{D : D(\mathbf{coeff}(f)) = 0,\ \forall f \in C\}$. Thus, $\mathcal{D}$ vanishes on C, but it also may vanish on other points. The *(Zariski) closure* of C, denoted $\overline{C}$, is simply all polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]^d$ which the distinguishers $\mathcal{D}$ vanish on, that is $\overline{C} = \{f \in \mathbb{F}[x_1, \ldots, x_n]^d : D(\mathbf{coeff}(f)) = 0,\ \forall D \in \mathcal{D}\}$. Clearly $C \subseteq \overline{C}$, but this is generally not an equality. For example, consider the map $(x, y) \mapsto (x, xy)$. It is easy to see that the image of this map is $\mathbb{F}^2 \setminus (\{0\} \times (\mathbb{F} \setminus \{0\}))$, but the closure is all of $\mathbb{F}^2$.

From the perspective of algebraic geometry, it is much more natural to study the closure $\overline{C}$ rather than the class C itself. And indeed, the algebraic natural proofs we define here necessarily give lower bounds for the closure $\overline{C}$ because the lower bound is proven using a distinguisher in $\mathcal{D}$. In fact, algebraic geometry shows that lower bounds for $\overline{C}$ *necessarily* must use such distinguishers (though they may not have small circuit size).[2] Thus, we see that

[2] It is unclear how much a difference this closure makes. For example, the exact relation between VP and $\overline{\mathsf{VP}}$ is unclear, see for example the work of Grochow, Mulmuley and Qiao [30]. It is conceivable that the algebraic distinguisher approach tries to prove too much, that is, perhaps $\overline{\mathsf{VP}} = \mathsf{VNP}$.

this distinguisher approach fits well into algebraic geometry and hence the GCT program.

Thus, the GCT approach fits into the algebraic natural proofs structure if one discards the (key) property of constructiveness. However, the GCT approach also uses more than just algebraic geometry and in particular relies on representation theory. That is, the GCT program notes that polynomials naturally have symmetries through linear changes of variables $\mathbf{x} \to A\mathbf{x}$ for an invertible matrix A and these symmetries act not only on the circuits C being computed but also their distinguishers $\mathcal{D}$. One can thus then ask that the lower bounds methods respect these symmetries, and Grochow [28] showed that most lower bounds in the literature do obey the natural symmetries one would expect. The goal of the GCT program is to use the symmetries of the distinguishers $\mathcal{D}$ to narrow down the search for them.

It is unclear to what extent constructivity plays a role in such arguments and as such the GCT program is not a-priori algebraically natural in the sense given here. Indeed, if there is an algebraically natural proofs barrier then the distinguishers that vanish on VP must have super-polynomial complexity, so that then clearly GCT is not constructive. This viewpoint demonstrates that our succinct hitting set constructions have relevance to GCT as they prove super-polynomial lower bounds for distinguishers that vanish on VP (also known as the defining equations), at least in the restricted models we consider.

2 PRELIMINARIES

We use boldface letters to denote vectors, where the length of a vector is usually understood from the context. Vectors such as $\mathbf{x}, \mathbf{y}$ and so on denote vectors of variables, where as $\boldsymbol{\alpha}, \boldsymbol{\beta}$ are used to denote vectors of scalars. As done in the introduction, we will express polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ in their monomial basis $f(\mathbf{x}) = \sum_{\mathbf{a}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}$ and then the corresponding vector of coefficients $\mathbf{coeff}(f) = (c_{\mathbf{a}})_{\mathbf{a}}$ can then be the input space to another polynomial $D \in \mathbb{F}[\{c_{\mathbf{a}}\}_{\mathbf{a}}]$. The exact size of this coefficient vector will be clear from context, that is, whether f is multilinear (so there are $N^{\text{ideg}}_{n,1} = 2^n$ coefficients) or whether f is of total degree at most d (so there are $N_{n,d} = \binom{n+d}{d}$ coefficients). Occasionally, we have a polynomial $f \in \mathbb{F}[\mathbf{x}, \mathbf{y}]$, and in that case we denote $\mathbf{coeff}_{\mathbf{x}}(f)$ the coefficient-vector of f where we think of $f \in (\mathbb{F}[\mathbf{y}])[\mathbf{x}]$, that is, the entries of the vector are now polynomials in $\mathbf{y}$.

For a polynomial f, we denote by $\|f\|_0$ the sparsity of f, i.e., the number of non-zero monomials appearing in f.

The proofs of most statements in the following sections are omitted from this version, and are available at the full version of this paper [23].

3 UNIVERSAL CONSTRUCTIONS OF PSEUDORANDOM POLYNOMIALS

In this section we detail *universal circuits* and their applications to pseudorandom polynomials. That is, a universal circuit for small computation is a polynomial $U(\mathbf{x}, \mathbf{y})$ such that for any polynomial $f(\mathbf{x})$ computed by a small computation, there is some value $\boldsymbol{\alpha}$ such that $f(\mathbf{x}) = U(\mathbf{x}, \boldsymbol{\alpha})$. Intuitively, there should be such universal circuits due to various completeness results, such as the fact that the determinant is complete for algebraic branching programs ([71]) (and hence complete for VP under quasipolynomial-size reductions ([72])). One would then expect that *if* there are pseudorandom polynomials then such universal circuits would also be pseudorandom.

Such construction was given by Raz [54], who embedded a generic *low-depth* computation, using that this are complete for VP due to the depth reduction of Valiant, Skyum, Berkowitz and Rackoff [72]. We now state this result.

Theorem 3.1 (Raz [54]). *Let $\mathbb{F}$ be a field, and let $n, s \geq 1$ and $d \geq 0$. Then there is a $\text{poly}(n, d, s)$-explicit $\text{poly}(n, d, s)$-size algebraic circuit $U \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_r]$ with $r \leq \text{poly}(n, d, s)$ such that*

- $\deg_{\mathbf{x}} U(\mathbf{x}, \mathbf{y}) \leq d$
- $\deg_{\mathbf{y}} U(\mathbf{x}, \mathbf{y}) \leq \text{poly}(d)$
- *If $f \in \mathbb{F}[\mathbf{x}]$ has $\deg_{\mathbf{x}} f \leq d$ then there is some $\boldsymbol{\alpha} \in \mathbb{F}^r$ such that $f(\mathbf{x}) = U(\mathbf{x}, \boldsymbol{\alpha})$.*

We briefly note that this construction also yields a universal circuit for homogeneous degree-d computations (that is, for the space $\mathbb{F}[x_1, \ldots, x_n]^d_{\text{hom}}$). No such universal circuits are known for efficient multilinear computation (the space $\mathbb{F}[x_1, \ldots, x_n]^1_{\text{ideg}}$), as circuits do not likely admit efficient multilinearization. In contrast, there is a universal circuit for the depth-3 set-multilinear formulas, which is the model that we use to construct our succinct hitting sets fooling restricted classes of computation. However, we restrict attention to total degree d polynomials as this is the cleanest setting.

We now use this universal circuit to convert from succinct hitting sets to succinct generators, as the standard conversion from hitting set to generator would ruin succinctness.

Lemma 3.2. *Let $\mathbb{F}$ be a field, and let $n, s \geq 1$ and $d \geq 0$. Let $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_{N_{n,d}}]$ be a class of polynomials in the coefficient vectors of $\mathbb{F}[x_1, \ldots, x_n]^d$. If there is an s-succinct hitting set for $\mathcal{D}$ then there is a $\text{poly}(n, d, s)$-succinct generator for $\mathcal{D}$ computable by $\text{poly}(n, d, s)$-size circuits.*

As mentioned in the introduction, generators are more robust versions of hitting sets. We now give another reason for this, by proving that succinct generators imply succinct hitting sets of *small size*, by using the standard interpolation argument.

Lemma 3.3. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| > \delta\Delta$, where $\Delta, \delta \geq 0$. Let $n, s \geq 1$ and $d \geq 0$. Let $\mathcal{D} \subseteq \mathbb{F}[c_1, \ldots, c_{N_{n,d}}]$ be a class of degree-Δ polynomials in the coefficient vectors of $\mathbb{F}[x_1, \ldots, x_n]^d$. Suppose that $G \in \mathbb{F}[\mathbf{x}, y_1, \ldots, y_\ell]$ is a succinct generator computable in size-s for $\mathcal{D}$ where $\deg_{\mathbf{y}} G \leq \delta$. Then there is a s-succinct hitting set of size $(\delta\Delta + 1)^\ell$.*

In the usual range of parameters we would have $\Delta = \text{poly}(N)$ and $\delta = \text{poly}(n, s)$. Plugging this into the above connections, we see that *any* (even infinite) succinct hitting set implies quasipolynomial-size hitting sets.

Corollary 3.4. *Let $\mathbb{F}$ be a field, and let $n \geq 1$. Consider polynomials in $\mathbb{F}[c_1, \ldots, c_N]$ where $N = \binom{2n}{n}$ so that $\mathbb{F}[c_1, \ldots, c_N]$ can be identified with the coefficients of polynomial sin $\mathbb{F}[x_1, \ldots, x_n]^d$ with $d = n$. If $\text{poly}(N)$-size $\text{poly}(N)$-degree circuits in $\mathbb{F}[c_1, \ldots, c_N]$ have $\text{poly}(n)$-succinct hitting sets from $\mathbb{F}[\mathbf{x}]^n$, then such circuits have an explicit $\text{poly}(N)^{\text{polylog } N}$-size hitting set.*

4 SUCCINCT HITTING SETS VIA RANK CONDENSERS

In this section, we construct succinct generators for restricted depth-3 formulas ($\Sigma\Pi\Sigma$ formulas), in particular, $\Sigma^k\Pi\Sigma$ formulas (top-fan-in k) and depth-3 circuits with bounded transcendence degree. The constructions are based on a common tool which we dub *succinct rank condenser*.

Gabizon and Raz [26], in the context of studying deterministic extractors, studied how to pseudorandomly map $\mathbb{F}^n$ to $\mathbb{F}^r$ preserving vector spaces of dimension r with high probability. In particular, they gave a poly(n)-collection of linear maps $\mathcal{E} = \{E : \mathbb{F}^n \to \mathbb{F}^r\}$ such that for any vector space $V \subseteq \mathbb{F}^n$ of dimension r there was at least one map $E \in \mathcal{E}$ such that the dimension of V was preserved, that is, $\dim E(V) = \dim V = r$. Their construction was improved by Forbes-Shpilka [21], and was called a *rank condenser* in later works ([19, 20]) which further explored this concept.

Rank condensers have proven very useful in designing hitting sets as they can reduce n-variate polynomials to r-variate polynomials, and for us the Gabizon and Raz [26] construction suffices. In particular, one defines the map $E \in \mathbb{F}[t]^{n\times r}$ with $E_{i,j} = t^{ij}$, with t is a formal variable. One can then obtain the desired collection $\mathcal{E}$ by evaluating $E(t)$ at sufficiently many points in $t \in \mathbb{F}$. However, it suffices for us to obtain generators, so we leave t as a formal variable.

Construction 4.1 (Succinct Rank Condenser). *Let $n \geq r \geq 1$. Define $P^{\mathrm{RC}}_{n,r} \in \mathbb{F}[x_1, \ldots, x_n, y_1, \ldots, y_r, t_0, t_1, \ldots, t_n]$ to be the polynomial*

$$P^{\mathrm{RC}}_{n,r}(\mathbf{x}, \mathbf{y}, \mathbf{t}) = \sum_{j=1}^{r} y_j t_0^j \prod_{k=1}^{n} (1 + x_k t_k^j) \ .$$

Let $\mathcal{G}^{\mathrm{RC}}_{n,r}(\mathbf{y}, \mathbf{t})$ be the polynomial map given by $\mathbf{coeff}_{\mathbf{x}}(P^{\mathrm{RC}}_{n,r})$ when taking $P^{\mathrm{RC}}_{n,r}$ as a multilinear polynomial in $\mathbf{x}$.

We now analyze properties of Construction 4.1, in particular showing that it embeds the desired rank condenser of Gabizon and Raz [26].

Proposition 4.2. *Assume the setup of Construction 4.1. Taking $N = 2^n$, identify $[N]$ with $2^{[n]}$. Then for every $i \in [N]$,*

$$\left(\mathcal{G}^{\mathrm{RC}}_{n,r}(\mathbf{x}, \mathbf{y}, t, t^{2^0}, t^{2^1} \cdots, t^{2^{n-1}})\right)_i = \sum_{j=1}^{r} y_j t^{ij}$$

We now observe that this generator is efficiently computable.

Proposition 4.3. *Assume the setup of Construction 4.1. The polynomial $P^{\mathrm{RC}}_{n,r}(\mathbf{x}, \mathbf{y}, \mathbf{t})$ is computable by* poly(n, r)*-size $\Sigma\Pi\Sigma\Pi$ circuits of* poly(n, r)*-degree. Further, for every fixing $\mathbf{y} = \boldsymbol{\alpha} \in \mathbb{F}^r$, $\mathbf{t} = \boldsymbol{\beta} \in \mathbb{F}^{n+1}$, $P^{\mathrm{RC}}_{n,r}(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$ is computed by a $\Sigma\Pi\Sigma$ circuit of size* poly(r, n).

In the full version of the paper, we use Construction 4.1 to obtain succinct generators for depth-3 formulas with bounded top-fan-in, and for depth-3 circuits of bounded transcendence degree.

5 SUCCINCT HITTING SETS VIA THE SHPILKA-VOLKOVICH GENERATOR

The Shpilka-Volkovich Generator (SV Generator, henceforth, and see [65]) is a polynomial map $\mathcal{G}(y_1, \ldots, y_k, z_1, \ldots, z_k) : \mathbb{F}^{2k} \to \mathbb{F}^N$ that satisfies the property that for every $T \subseteq [N]$ such that $|T| \leq k$, we can set $z_1, \ldots, z_k$ to values $\alpha_{i_1}, \ldots, \alpha_{i_k}$ such that the $\mathbf{y}$ variables are mapped to the locations indexed by T, and the other coordinates of the polynomial map are zeroed out. This property turns out to be immensely useful in constructing hitting sets for various classes. Hence, we begin by constructing a succinct analog of this generator, and then use it to obtain succinct hitting sets in cases where the SV generator is applicable.

Construction 5.1 (Succinct SV Generator). *Let $n \in \mathbb{N}$ and $N = 2^n$. Define*

$$P(z_1, \ldots, z_n, x_1, \ldots, x_n) = \prod_{i=1}^{n} (z_i \cdot x_i + (1 - z_i)),$$

and

$$Q^{\mathrm{SSV}}_{n,k}(y_1, \ldots, y_k, z_{1,1}, \ldots, z_{1,n}, \ldots, z_{k,1}, \ldots, z_{k,n}, x_1, \ldots, x_n) = \sum_{i\in[k]} y_i \cdot P(\mathbf{z}_i, \mathbf{x}),$$

where $\mathbf{z}_i = (z_{i,1}, \ldots, z_{i,n})$. Finally, let

$$\mathcal{G}^{\mathrm{SSV}}_{n,k}(\mathbf{z}_1, \ldots, \mathbf{z}_k, y_1, \ldots, y_k) = \mathbf{coeff}_{\mathbf{x}}(Q^{\mathrm{SSV}}_{n,k}(\mathbf{y}, \mathbf{z}, \mathbf{x})).$$

We begin by stating an immediate fact regarding Construction 5.1.

Fact 5.2 (Succinctness). *For every setting $\mathbf{y} = \alpha, \mathbf{z} = \beta$, the polynomial $Q^{\mathrm{SSV}}_{n,k}$ is computed by a multilinear $\Sigma\Pi\Sigma$ circuit of size* poly(n, k).

The usefulness of the generator comes from the following property, which is, in some sense, the algebraic analog of k-wise independence.

Lemma 5.3. *For every $T \subseteq [N]$ such that $|T| \leq k$, there is a fixing of the $\mathbf{z}$ variables, and possibly of some of the $\mathbf{y}$ variables, such that in the mapping $\mathcal{G}^{\mathrm{SSV}}_{n,k}$, $|T|$ distinct $\mathbf{y}$ variables are planted in the coordinates corresponding to T, while the rest of the entries are zeroed out.*

In the full version of this paper, we use Construction 5.1 and some variants thereof to obtain succinct generators for sparse polynomials, sums of powers of low degree polynomials, commutative read-once algebraic branching programs and depth-D occur-k formulas, and circuits of small transcendence degree.

6 SUCCINCT HITTING SETS FOR READ-ONCE OBLIVIOUS ALGEBRAIC BRANCHING PROGRAMS

In this section we construct a succinct hitting set for the class of read-once oblivious algebraic programs. Recall that in Section 5 we have constructed a poly$(\log w, \log n)$-$\Sigma\Pi\Sigma$ succinct generator for width-w *commutative* roABPs. For general ABPs, we are only able at this point to construct hitting sets that are width-w^2 roABP succinct: i.e., in the hitting set for width w N-variate roABPs, each element is computed by a width w^2 n-variate roABP. Ideally, one would want to replace w^2 with polylog(w).

The definition of roABPs are given in the full version of this paper. Throughout this section we assume that the ABP reads the

variables in the order $X_1, X_2, \ldots, X_N$. In Section 6.1 we give some short remarks regarding different variable orderings.

Our construction is based on the following generator by Forbes and Shpilka [22].

Lemma 6.1 (Forbes-Shpilka Generator for roABPs, Construction 3.13 in [22]). *Let $n \in \mathbb{N}$ and $N = 2^n$. The following polynomial map $\mathcal{G} : \mathbb{F}^{n+1} \to \mathbb{F}^N$ is a generator for width w, individual degree d, N-variate roABPs, in variable order $X_1, X_2, \ldots, X_N$.*

Let $\omega \in \mathbb{F}$ be of multiplicative order at least $(Ndw^2)^2$, and let $\beta_1, \ldots, \beta_{w^2}$ be distinct elements of $\mathbb{F}$. Let $\left\{p_\ell : \ell \in [w^2]\right\}$ be the Lagrange interpolation polynomials with respect to the β_i's, i.e., $p_i(\beta_j) = 1$ if $i = j$ and 0 otherwise.

Let $\mathcal{G} : \mathbb{F}^{n+1} \to \mathbb{F}^N$ be the following polynomial map, whose output coordinates are indexed by vectors $\mathbf{b} \in \{0,1\}^n$.

$$\mathcal{G}^{\mathrm{FS}}_{\mathbf{b}}(\mathbf{y}) = \sum_{\ell_1,\ldots,\ell_n \in [w^2]} \prod_{i \in [n]} \Big((1-b_i) \cdot p_{\ell_{i-1}}(\omega^{\ell_i} y_i) + b_i \cdot p_{\ell_{i-1}}((\omega^{\ell_i} y_i)^{2^{i-1} d w^2}) \Big) \cdot p_{\ell_n}(y_{n+1}), \tag{1}$$

where we abuse notation by defining $p_{\ell_0}(t) = t$.

In [22] (Lemma 3.18), it is shown that this map, for every fixed output coordinate $\mathbf{b}$, is computed by a width w^2 roABP in the variables $\mathbf{y}$. We, however, want to show that for every fixing $\mathbf{y} = \boldsymbol{\alpha}$, there is a small roABP computing the polynomial whose coefficient vector is given by $(\mathcal{G}_{\mathbf{b}}(\boldsymbol{\alpha}))_{\mathbf{b} \in \{0,1\}^n}$. That is, for every choice of $\boldsymbol{\alpha}$, and associating $\mathbf{b}$ with a subset of $[n]$, we want a polynomial in $x_1, \ldots, x_n$ such that the coefficient of $\mathbf{x}_b$ is $\mathcal{G}_{\mathbf{b}}(\boldsymbol{\alpha})$.

Definition 6.2 (Succinct Forbes-Shpilka Generator). Let $n, w \in \mathbb{N}$, and ω, p_i's as in Lemma 6.1. Define

$$P^{\mathrm{FS}}(x_1, \ldots, x_n, y_1, \ldots, y_{n+1}) = \sum_{\ell_1,\ldots,\ell_n \in [w^2]} \prod_{i \in [n]} \Big(p_{\ell_{i-1}}(\omega^{\ell_i} y_i) + x_i \cdot p_{\ell_{i-1}}((\omega^{\ell_i} y_i)^{2^{i-1} d w^2}) \Big) \cdot p_{\ell_n}(y_{n+1})$$

We first claim the the Forbes-Shpilka generator (1) is given by the coefficient vector of this polynomial.

Claim 6.3. *Assume the setup and notations of Definition 6.2. Then $\mathbf{coeff}_{\mathbf{x}}(P^{\mathrm{FS}}) = \mathcal{G}^{\mathrm{FS}}$.*

We now show that for every fixing $\mathbf{y} = \boldsymbol{\alpha}$, the polynomial $P^{\mathrm{FS}}(\mathbf{x}, \boldsymbol{\alpha})$ is computed by a small roABP.

Claim 6.4. *For every setting $\mathbf{y} = \boldsymbol{\alpha}$, the polynomial $P^{\mathrm{FS}}(\mathbf{x}, \boldsymbol{\alpha})$ in Definition 6.2 can be computed by a width w^2 roABP in variable order $x_1, x_2 \ldots, x_n$.*

Corollary 6.5. *The Forbes-Shpilka generator given in Lemma 6.1 is a width w^2-roABP succinct generator for degree d roABPs that read the variables in order $X_1, X_2, \ldots, X_N$.*

6.1 Different Variable Orderings

The generator given by Forbes and Shpilka in Lemma 6.1 hits roABPs that read the variables in the order $X_1, X_2, \ldots, X_N$ and not necessarily in any variable order. Obviously, we can apply a permutation σ to the variables $x_1, \ldots, x_n$ in Definition 6.2 to obtain a roABP in the variables $\mathbf{x}$ in the order σ: the coefficient vector of this roABP hits roABPs in the variables $\mathbf{X}$ that read their variables in the order on $\{X_1, \ldots, X_N\}$ which is given by considering the lexicographic ordering induced on the set of multilinear monomials in $\{x_1, \ldots, x_n\}$ by the order σ, and using the canonical identification of a multilinear monomial with an index in $[N]$, say, using the binary representation. We call such an order relation on $[N]$ a *monomial-compatible* ordering. Note that there are merely $n!$ such orderings among the $N!$ total orderings on $[N]$.

Since in our case we do not care about the *size* of the hitting set, we can take the union of all $n!$ those succinct hitting sets to obtain the following corollary.

Corollary 6.6. *There exists a width-w^2 roABP succinct hitting set for the class of width w, N variate, and degree d roABPs that read the variables in a monomial compatible ordering.*

7 DISCUSSION AND OPEN PROBLEMS

In this work, we have shown that many of the hitting sets we know for restricted algebraic models of computation can be represented in a succinct form as coefficient vectors of small circuits. This gives some positive answers to Conjecture 1.9, and points to the possibility of an algebraic natural proofs barrier. The main problem left open by this work is to construct succinct hitting sets for stronger models for which we know how to construct hitting sets efficiently.

For example, while we were able to construct a succinct generator for commutative roABPs, our construction for general roABPs is not fully succinct, and also works only in certain variable orderings. Despite several works that obtain quasi-polynomial size hitting sets for roABPs in any order ([5, 20]), none of them seems to fit easily into the succinct setting, each for its own reasons.

The main technical tool which we do not know how to emulate in the succinct setting is the Klivans-Spielman [42] generator. In this generator, the variable X_i is mapped to $t^{k^i \bmod p}$, where t is a new indeterminate, p is chosen from an appropriately large set of primes and k from an appropriately large set of natural numbers. The main feature of this generator is that given a "small" enough set of monomials $\mathcal{M}$, the parameters k, p can be chosen from a "not too large" set, such that all the monomials in $\mathcal{M}$ are given distinct weights, and this can be done in a black-box manner, that is, without knowing $\mathcal{M}$, but only an upper bound on its size.

The main application of the Klivans and Spielman construction is to construct hitting sets for sparse polynomials. While we are unable to make the resulting hitting set succinct, we developed an alternate hitting set which we succeeding in making succinct. However, the Klivans and Spielman construction (or otherwise similar ideas) has also found applications beyond the class of sparse polynomials, such as in the construction hitting sets for roABPs in unknown order from the work of Agrawal, Gurjar, Korwar and Saxena [5]. Unfortunately, such works seem to rely heavily on properties of the Klivans and Spielman construction beyond that of just hitting sparse polynomials, and as such we are currently unable to make these hitting sets succinct.

A particular interesting application of the Klivans and Spielman construction is in the recent works of Fenner, Gurjar and Thierauf [18] and its generalization by Gurjar and Thierauf [33]. These

works construct hitting sets for the class of determinants of "read-once matrices", which are polynomials of the form $\det M$, where M is a matrix in which each entry contains a variable $x_{i,j}$ or a field constant, and each variable appears at most once in the matrix. While this class of polynomials is very restricted, the partial derivative matrix used by Nisan [50], Raz [53], and Raz-Yehudayoff [56], is a read-once matrix. As such, the lower bounds proved in these papers are algebraically natural and the distinguisher used is a read-once determinant. The work of Raz and Yehudayoff [56] in particular shows that a read-once determinant can vanish on the coefficient vectors of constant-depth multilinear formulas, and as most of the constructions in this paper have this form this shows that these constructions cannot be succinct hitting sets for read-once determinants, and hence new ideas are needed. Indeed, if one could establish a circuit class C where there are C-succinct hitting sets for read-once determinants than this would show that no proof technique following the ideas of the above works can prove lower bounds for the class C. Such a result would be very interesting as those lower bounds methods are still very much state-of-the-art.

As mentioned earlier, stronger evidence towards an algebraic natural proofs barrier can also be obtained by designing pseudo-random polynomials whose security is based on widely-believed cryptographic assumptions. In particular, one possible approach is obtaining evidence in favor of the determinant-based construction of Aaronson and Drucker [2].

ACKNOWLEDGMENTS

We thank Scott Aaronson, Andy Drucker, Josh Grochow, Mrinal Kumar, Shubhangi Saraf and Dor Minzer for useful conversations regarding this work.

REFERENCES

[1] Scott Aaronson. 2016. P $\stackrel{?}{=}$ NP. In *Open Problems in Mathematics*. Springer, 1–122. http://www.scottaaronson.com/papers/pnp.pdf

[2] Scott Aaronson and Andrew Drucker. 2008. Arithmetic natural proofs theory is sought. (2008). http://www.scottaaronson.com/blog/?p=336 Blog post, http://www.scottaaronson.com/blog/?p=336.

[3] Scott Aaronson and Avi Wigderson. 2009. Algebrization: A New Barrier in Complexity Theory. *TOCT* 1, 1 (2009), 2:1–2:54. DOI: http://dx.doi.org/10.1145/1490270.1490272

[4] Manindra Agrawal. 2005. Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*. 92–105. DOI: http://dx.doi.org/10.1007/11590156_6

[5] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. 2015. Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM J. Comput.* 44, 3 (2015), 669–697. DOI: http://dx.doi.org/10.1137/140975103 arXiv:1406.7535

[6] Manindra Agrawal and V. Vinay. 2008. Arithmetic Circuits: A Chasm at Depth Four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*. 67–75. DOI: http://dx.doi.org/10.1109/FOCS.2008.32

[7] Miklós Ajtai. 1983. Σ^1_1-formulae on finite structures. *Annals of pure and applied logic* 24, 1 (1983), 1–48. DOI: http://dx.doi.org/10.1016/0168-0072(83)90038-6

[8] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. 1998. Non-Commutative Arithmetic Circuits: Depth Reduction and Size Lower Bounds. *Theoretical Computer Science* 209, 1-2 (1998), 47–86. DOI: http://dx.doi.org/10.1016/S0304-3975(97)00227-2

[9] Noga Alon and Ravi B. Boppana. 1987. The monotone circuit complexity of Boolean functions. *Combinatorica* 7, 1 (1987), 1–22. DOI: http://dx.doi.org/10.1007/BF02579196

[10] Theodore P. Baker, John Gill, and Robert Solovay. 1975. Relativizations of the P =? NP Question. *SIAM J. Comput.* 4, 4 (1975), 431–442. DOI: http://dx.doi.org/10.1137/0204037

[11] Walter Baur and Volker Strassen. 1983. The Complexity of Partial Derivatives. *Theoretical Computer Science* 22 (1983), 317–330. DOI: http://dx.doi.org/10.1016/0304-3975(83)90110-X

[12] Stuart J. Berkowitz. 1984. On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.* 18, 3 (1984), 147 – 150. DOI: http://dx.doi.org/10.1016/0020-0190(84)90018-8

[13] Markus Bläser. 2014. Explicit tensors. In *Perspectives in Computational Complexity*. Springer, 117–130. DOI: http://dx.doi.org/10.1007/978-3-319-05446-9_6

[14] Peter Bürgisser, Michael Clausen, and Mohammad A. Shokrollahi. 1997. *Algebraic Complexity Theory*. Grundlehren der mathematischen Wissenschaften, Vol. 315. Springer-Verlag. DOI: http://dx.doi.org/10.1007/978-3-662-03338-8

[15] Suryajith Chillara, Mrinal Kumar, Ramprasad Saptharishi, and V. Vinay. 2016. The Chasm at Depth Four, and Tensor Rank : Old results, new insights. *Electronic Colloquium on Computational Complexity (ECCC)* 23 (2016), 96. http://eccc.hpi-web.de/report/2016/096

[16] Richard A. DeMillo and Richard J. Lipton. 1978. A Probabilistic Remark on Algebraic Program Testing. *Inform. Process. Lett.* 7, 4 (1978), 193–195. DOI: http://dx.doi.org/10.1016/0020-0190(78)90067-4

[17] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. 2009. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM J. Comput.* 39, 4 (2009), 1279–1293. DOI: http://dx.doi.org/10.1137/080735850

[18] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. 2016. Bipartite perfect matching is in quasi-NC. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing (STOC 2016)*. ACM, 754–763. DOI: http://dx.doi.org/10.1145/2897518.2897564

[19] Michael A. Forbes and Venkatesan Guruswami. 2015. Dimension Expanders via Rank Condensers. In *Proceedings of the 19th International Workshop on Randomization and Computation (RANDOM 2015) (LIPIcs)*, Vol. 40. 800–814. DOI: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.800 Full version at arXiv:1411.7455.

[20] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. 2014. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*. 867–875. DOI: http://dx.doi.org/10.1145/2591796.2591816

[21] Michael A. Forbes and Amir Shpilka. 2012. On Identity Testing of Tensors, Low-rank Recovery and Compressed Sensing. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*. 163–172. DOI: http://dx.doi.org/10.1145/2213977.2213995 Full version at arXiv:1111.0663.

[22] Michael A. Forbes and Amir Shpilka. 2013. Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*. 243–252. DOI: http://dx.doi.org/10.1109/FOCS.2013.34 Full version at arXiv:1209.2408.

[23] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. 2017. Succinct Hitting Sets and Barriers to Proving Algebraic Circuits Lower Bounds. *Electronic Colloquium on Computational Complexity (ECCC)* 24 (2017), 7. https://eccc.weizmann.ac.il/report/2017/007

[24] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. 2014. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*. 128–135. http://doi.acm.org/10.1145/2591796.2591824

[25] Merrick L. Furst, James B. Saxe, and Michael Sipser. 1984. Parity, Circuits, and the Polynomial-Time Hierarchy. *Mathematical Systems Theory* 17, 1 (1984), 13–27. DOI: http://dx.doi.org/10.1007/BF01744431

[26] Ariel Gabizon and Ran Raz. 2008. Deterministic extractors for affine sources over large fields. *Combinatorica* 28, 4 (2008), 415–440. DOI: http://dx.doi.org/10.1007/s00493-008-2259-3 Preliminary version in the *46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*.

[27] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. 1986. How to construct random functions. *J. ACM* 33, 4 (1986), 792–807. DOI: http://dx.doi.org/10.1145/6490.6503

[28] Joshua A. Grochow. 2015. Unifying Known Lower Bounds via Geometric Complexity Theory. *Computational Complexity* 24, 2 (2015), 393–475. DOI: http://dx.doi.org/10.1007/s00037-015-0103-x Preliminary version in the *29th Annual IEEE Conference on Computational Complexity (CCC 2014)*.

[29] Joshua A. Grochow, Mrinal Kumar, Michael Saks, and Shubhangi Saraf. 2017. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR* abs/1701.01717 (2017). https://arxiv.org/abs/1701.01717

[30] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. 2016. Boundaries of VP and VNP. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016) (LIPIcs)*, Vol. 55. 34:1–34:14. DOI: http://dx.doi.org/10.4230/LIPIcs.ICALP.2016.34 Full version at arXiv:abs/1605.02815.

[31] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2014. Approaching the Chasm at Depth Four. *J. ACM* 61, 6 (2014), 33:1–33:16. DOI: http://dx.doi.org/10.1145/2629541

[32] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2016. Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.* 45, 3 (2016), 1064–1079. DOI: http://dx.doi.org/10.1137/140957123

[33] Rohit Gurjar and Thomas Thierauf. 2016. Linear Matroid Intersection is in quasi-NC. *Electronic Colloquium on Computational Complexity (ECCC)* 23 (2016), 182. http://eccc.hpi-web.de/report/2016/182

[34] Juris Hartmanis and Richard E. Stearns. 1965. On the Computational Complexity of Algorithms. *Trans. Amer. Math. Soc.* 117 (1965), 285–306. DOI: http://dx.doi.org/10.2307/1994208

[35] Johan Håstad. 1989. Almost Optimal Lower Bounds for Small Depth Circuits. In *Randomness and Computation*. JAI Press, 6–20. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.100.8571

[36] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. 1999. A Pseudorandom Generator from any One-way Function. *SIAM J. Comput.* 28, 4 (1999), 1364–1396. DOI: http://dx.doi.org/10.1137/S0097539793244708

[37] Joos Heintz and Claus-Peter Schnorr. 1980. Testing Polynomials which Are Easy to Compute (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*. 262–272. DOI: http://dx.doi.org/10.1145/800141.804674

[38] Valentine Kabanets and Russell Impagliazzo. 2004. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity* 13, 1-2 (2004), 1–46. DOI: http://dx.doi.org/10.1007/s00037-004-0182-6 Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[39] Neeraj Kayal. 2012. An exponential lower bound for the sum of powers of bounded degree polynomials. In *Electronic Colloquium on Computational Complexity (ECCC)TR12-081*. http://eccc.hpi-web.de/report/2012/081

[40] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. 2014. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Circuits. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*. 61–70. DOI: http://dx.doi.org/10.1109/FOCS.2014.15

[41] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. 2014. A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*. 146–153. http://doi.acm.org/10.1145/2591796.2591847

[42] Adam Klivans and Daniel A. Spielman. 2001. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*. 216–223. DOI: http://dx.doi.org/10.1145/380752.380801

[43] Pascal Koiran. 2012. Arithmetic Circuits: The Chasm at Depth Four Gets Wider. *Theoretical Computer Science* 448 (2012), 56–65. DOI: http://dx.doi.org/10.1016/j.tcs.2012.03.041 arXiv:1006.4700

[44] Matthias Krause and Stefan Lucks. 2001. Pseudorandom functions in TC^0 and cryptographic limitations to proving lower bounds. *Computational Complexity* 10, 4 (2001), 297–313. DOI: http://dx.doi.org/10.1007/s000370100002

[45] Mrinal Kumar and Shubhangi Saraf. 2014. On the power of homogeneous depth 4 arithmetic circuits. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*. 364–373. DOI: http://dx.doi.org/10.1109/FOCS.2014.46

[46] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. 1992. Algebraic Methods for Interactive Proof Systems. *J. ACM* 39, 4 (1992), 859–868. DOI: http://dx.doi.org/10.1145/146585.146605

[47] Meena Mahajan and V. Vinay. 1997. A Combinatorial Algorithm for the Determinant. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 1997)*. 730–738. DOI: http://dx.doi.org/10.1.1.31.1673 Available on `citeseer:10.1.1.31.1673`.

[48] Ketan D Mulmuley and Milind Sohoni. 2001. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM J. Comput.* 31, 2 (2001), 496–526. DOI: http://dx.doi.org/10.1137/S009753970038715X

[49] Moni Naor and Omer Reingold. 1997. Number-theoretic Constructions of Efficient Pseudo-random Functions. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1997)*. IEEE Computer Society, 458–467. DOI: http://dx.doi.org/10.1109/SFCS.1997.646134

[50] Noam Nisan. 1991. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*. 410–418. DOI: http://dx.doi.org/10.1145/103418.103462 Available on `citeseer:10.1.1.17.5067`.

[51] Noam Nisan and Avi Wigderson. 1997. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity* 6, 3 (1997), 217–234. DOI: http://dx.doi.org/10.1007/BF01294256 Available on `citeseer:10.1.1.90.2644`.

[52] Ran Raz. 2006. Separation of Multilinear Circuit and Formula Size. *Theory of Computing* 2, 1 (2006), 121–135. DOI: http://dx.doi.org/10.4086/toc.2006.v002a006 Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*.

[53] Ran Raz. 2009. Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size. *J. ACM* 56, 2 (2009), 8:1–8:17. DOI: http://dx.doi.org/10.1145/1502793.1502797 Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*.

[54] Ran Raz. 2010. Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory of Computing* 6, 1 (2010), 135–177. DOI: http://dx.doi.org/10.4086/toc.2010.v006a007

[55] Ran Raz and Amir Yehudayoff. 2008. Balancing Syntactically Multilinear Arithmetic Circuits. *Computational Complexity* 17, 4 (2008), 515–535. DOI: http://dx.doi.org/10.1007/s00037-008-0254-0

[56] Ran Raz and Amir Yehudayoff. 2009. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity* 18, 2 (2009), 171–207. DOI: http://dx.doi.org/10.1007/s00037-009-0270-8 Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[57] Alexander A Razborov. 1985. Lower bounds on the monotone complexity of some Boolean functions. In *Dokl. Akad. Nauk SSSR*, Vol. 281(4). 798–801. http://people.cs.uchicago.edu/~razborov/files/clique.pdf Translation in Soviet Math. Doklady, 31, 354–357.

[58] Alexander A. Razborov. 1987. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR* 41, 4 (1987), 333–338. DOI: http://dx.doi.org/10.1007/BF01137685

[59] Alexander A. Razborov and Steven Rudich. 1997. Natural Proofs. *J. Comput. Syst. Sci.* 55, 1 (1997), 24–35. DOI: http://dx.doi.org/10.1006/jcss.1997.1494

[60] Ramprasad Saptharishi. 2016. A survey of lower bounds in arithmetic circuit complexity. (2016). https://github.com/dasarpmar/lowerbounds-survey/releases/ Github survey, https://github.com/dasarpmar/lowerbounds-survey/.

[61] Nitin Saxena. 2009. Progress on Polynomial Identity Testing. *Bulletin of the EATCS* 99 (2009), 49–79. http://eccc.hpi-web.de/report/2009/101/

[62] Nitin Saxena. 2014. Progress on Polynomial Identity Testing - II. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*. 131–146. DOI: http://dx.doi.org/10.1007/978-3-319-05446-9_7

[63] Jacob T. Schwartz. 1980. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* 27, 4 (1980), 701–717. DOI: http://dx.doi.org/10.1145/322217.322225

[64] Adi Shamir. 1990. IP=PSPACE. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science (FOCS 1990)*. 11–15. DOI: http://dx.doi.org/10.1109/FSCS.1990.89519

[65] Amir Shpilka and Ilya Volkovich. 2015. Read-once polynomial identity testing. *Computational Complexity* 24, 3 (2015), 477–532. DOI: http://dx.doi.org/10.1007/s00037-015-0105-8 Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*.

[66] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science* 5 (March 2010), 207–388. Issue 3–4. DOI: http://dx.doi.org/10.1561/0400000039

[67] Roman Smolensky. 1987. Algebraic Methods in the Theory of Lower Bounds for Boolean Circuit Complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987)*. 77–82. DOI: http://dx.doi.org/10.1145/28395.28404

[68] Volker Strassen. 1973. Die Berechnungskomplexität Von Elementarsymmetrischen Funktionen Und Von Interpolationskoeffizienten. *Numer. Math.* 20, 3 (June 1973), 238–251. DOI: http://dx.doi.org/10.1007/BF01436566

[69] Éva Tardos. 1988. The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica* 8, 1 (1988), 141–142. DOI: http://dx.doi.org/10.1007/BF02122563

[70] Sébastien Tavenas. 2015. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.* 240 (2015), 2–11. DOI: http://dx.doi.org/10.1016/j.ic.2014.09.004 Preliminary version in the *38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*.

[71] Leslie G. Valiant. 1979. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)*. 249–261. DOI: http://dx.doi.org/10.1145/800135.804419

[72] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. 1983. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.* 12, 4 (1983), 641–644. DOI: http://dx.doi.org/10.1137/0212043 Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*.

[73] R. Ryan Williams. 2016. Natural Proofs versus Derandomization. *SIAM J. Comput.* 45, 2 (2016), 497–529. DOI: http://dx.doi.org/10.1137/130938219

[74] Andrew Chi-Chih Yao. 1985. Separating the Polynomial-Time Hierarchy by Oracles (Preliminary Version). In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1985)*. 1–10. DOI: http://dx.doi.org/10.1109/SFCS.1985.49

[75] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation (Lecture Notes in Computer Science)*, Vol. 72. Springer, 216–226. DOI: http://dx.doi.org/10.1007/3-540-09519-5_73