THEORY OF COMPUTING, Volume 14 (18), 2018, pp. 1–45
www.theoryofcomputing.org

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

Michael A. Forbes* Amir Shpilka† Ben Lee Volk‡

Received July 16, 2017; Revised July 26, 2018; Published December 18, 2018

Abstract: We formalize a framework of *algebraically natural* lower bounds for algebraic circuits. Just as with the natural proofs notion of Razborov and Rudich (1997) for Boolean circuit lower bounds, our notion of algebraically natural lower bounds captures nearly all lower bound techniques known. However, unlike in 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 (2016) 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, to the existence of a hitting set for the class of $\mathrm{poly}(N)$-degree $\mathrm{poly}(N)$-size circuits which consists of coefficient vectors of polynomials of $\mathrm{polylog}(N)$ degree with $\mathrm{polylog}(N)$-size circuits. Further, we give

A conference version of this paper appeared in the Proceedings of the 49th Annual ACM Symposium on Theory of Computing (STOC 2017) [39].

*Supported by the NSF, including NSF CCF-1617580, and the DARPA Safeware program.

†Supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575, and by the Israel Science Foundation (grant number 552/16).

‡Supported by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575, and by the Israel Science Foundation (grant number 552/16).

ACM Classification: F.1.3

AMS Classification: 68Q15, 68Q17

Key words and phrases: algebraic circuits, lower bounds, derandomization, polynomial identity testing, barriers

 DOI: 10.4086/toc.2018.v014a018

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 (2001), 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.

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 $\mathsf{P} \neq \mathsf{EXP}$ ([52]), circuit lower bounds showing that $\mathsf{AC}^0 \neq \mathsf{P}$ ([9, 41, 106, 53]), and interactive proofs showing $\mathsf{IP} = \mathsf{PSPACE}$ ([67, 95]). 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. The above results are covered by the barriers of relativization of Baker, Gill and Solovay [14], natural proofs of Razborov and Rudich [86], and algebrization of Aaronson and Wigderson [3], respectively. In this paper we revisit the natural proofs barrier of Razborov and Rudich [86] and seek to understand how it extends to a barrier to algebraic circuit lower bounds. While previous work has considered versions of an algebraic natural proofs barrier ([2, 97, 44]), we give the *first* evidence of such a barrier against restricted algebraic reasoning.

Natural proofs. The setting of Razborov and Rudich [86] 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 as Turing machines, and can even (trivially) compute undecidable languages, their ability to solve computational problems of interest can seem closer to uniform computation. For example, if polynomial-size circuits can solve NP-hard problems then there are unexpected implications for uniform computation similar to $\mathsf{P} = \mathsf{NP}$ (the polynomial-time hierarchy collapses to $\Sigma_2^P = \Pi_2^P$ ([58])). Therefore, 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 [14] which seems to obstruct most methods that can exploit uniformity.

There have been many important lower bounds obtained for restricted classes of circuits: constant-depth circuits ([9, 41, 106, 53]), constant-depth circuits with prime modular gates ([85, 98]), as well as lower bounds for monotone circuits ([84, 13, 11, 100]). Razborov and Rudich [86] 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

$$P \subseteq \bigcup_{n \geq 1} \{f : \{0,1\}^n \to \{0,1\}\}$$

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 $\operatorname{poly}(n)$-size circuits then f has property P.
2. *Largeness:* With noticeable probability (of at least $1/\operatorname{poly}(N) = 2^{-O(n)}$), random functions $f : \{0,1\}^n \to \{0,1\}$ do not have the property P.
3. *Constructivity:* Given a truth-table of a 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 [86] showed that (possibly after a small modification) most circuit lower bounds (such as those for constant-depth circuits ([9, 41, 106, 53, 85, 98])) yield large and constructive properties, and called such lower bounds *natural proofs*.

Further, Razborov and Rudich [86] 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 *pseudorandom function* is an efficiently computable function $f : \{0,1\}^n \times \{0,1\}^\lambda \to \{0,1\}$ such that when sampling the key $\mathsf{k} \in \{0,1\}^\lambda$ at random, the resulting distribution of functions $f(\cdot,\mathsf{k})$ is computationally indistinguishable from a truly random function $\mathsf{f} : \{0,1\}^n \to \{0,1\}$. The existence of pseudorandom functions follows from the existence of *one-way functions* ([54, 43]) 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 [73], whose security rests on the intractability of discrete-log and factoring-type assumptions (see also Krause and Lucks [65]). It is therefore widely believed that there are pseudorandom functions, even ones computationally indistinguishable from random except to adversaries running in $\exp(\lambda^{\Omega(1)})$-time.

In contrast, Razborov and Rudich [86] showed that a natural proof useful against $\operatorname{poly}(n)$-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. Indeed, 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 $\operatorname{poly}(n)$-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 $f(\cdot,\mathsf{k})$ from a random function, as desired.

[1] The Razborov and Rudich [86] definition of a natural property actually applies to the complement of the property P we use here. In particular, our use of the term "largeness" refers to the fact that the complement of P is a large set. This is a trivial difference for Boolean complexity, but is important for algebraic complexity as there natural properties are one-sided, see Section 1.2.

While the natural proofs barrier has proved difficult to overcome, there are results that seem to circumvent it. For example, the barrier does not seem to apply to the lower bounds obtained for monotone circuits ([84]), as there the notion of a "random monotone function" is not well-defined. Further, there are results (such as Williams' [104] result of $\mathsf{ACC}^0 \neq \mathsf{NEXP}$) that circumvent the natural proofs barrier by incorporating techniques from uniform complexity. Other work has demonstrated that relaxing the notion of natural proof can avoid the implications to breaking cryptography. Chow [27] has shown that *almost* natural proofs (which relax largeness slightly) *can* prove super-polynomial circuit lower bounds (under plausible cryptographic or complexity-theoretic assumptions). Williams [105] has shown, among other results, that some circuit lower bounds (such as for EXP or NEXP) are *equivalent* to constructive (non-trivial) properties useful against small circuits, which yet have no need for any sort of largeness. Chapman and Williams [25] have shown that obtaining circuit lower bounds for a self-checkable problem (such as SAT) is *essentially equivalent* to obtaining a natural property against circuits that "check their work." These works suggest that the exact implications of the natural proofs barrier remains not fully understood.

Algebraic natural proofs. Algebraic circuits are one of the most natural models 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 [44] noticed that many of the prominent algebraic circuit lower bounds (such as [74, 77, 79, 83]) are *algebraically natural*, in that they obey an algebraic form of usefulness, largeness, and constructivity.

While this would seemingly then imply a Razborov and Rudich [86]-type barrier for existing techniques, there is a key piece missing: we have very little evidence for the existence of *algebraic* pseudorandom functions. The pseudorandom functions used by Razborov and Rudich [86] 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 sufficiently strong so that they can compute "enough" cryptography to be secure against algebraic circuits, so that a natural proofs barrier *would* apply.

Our results. In this paper we formalize the study of pseudorandom polynomials by exhibiting the *first* constructions provably secure against restricted classes of algebraic circuits. Our notion of pseudorandomness is related to the *polynomial identity testing* problem, the derandomization of which is one of the main open problems in algebraic complexity theory (see Section 1.3 for more details). In particular, we follow Williams [105] 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.

A more formal statement of the results appears in Section 1.5. In order to present them, however, we require some technical background and definitions, which will be presented in the forthcoming sections.

Recently, and independently of our work, Grochow, Kumar, Saks, and Saraf [45] observed a similar connection between a natural proofs barrier for algebraic circuits and succinct derandomization. They also present connections with Geometric Complexity Theory (which we discuss below in Section 1.7) and algebraic proof complexity. However, unlike our paper, 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 ($\times$). 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 to prove a super-polynomial lower bound for the algebraic circuit size of any explicit polynomial, that is, proving $\mathsf{VP} \neq \mathsf{VNP}$. 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 ([99, 15])—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 ([74]), multilinear formulas ([80, 82]), and homogeneous depth-3 and depth-4 circuits ([77, 48, 61, 40, 60, 66]). We refer the reader to Saptharishi [88] for a continuously updating comprehensive compendium of these lower bounds.

However, this landscape might still 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 ([103, 10, 8, 64, 101, 48, 26]) 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 ([59, 47, 61, 40, 60, 66]) 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 low-complexity and high-complexity polynomials, so that they are *useful* in the sense of Razborov and Rudich [86]. In particular, we want to know if such distinguishers[2] can be *efficient*, so that they are also *constructive*. Several works, such as Aaronson and Drucker [2], Grochow [44] (see also Shpilka and Yehudayoff [97, Section 3.9], and Aaronson [1, Section 6.5.3]) have noticed that almost 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 ([77]), evaluation/coefficient dimension ([74, 79, 83, 37]), or shifted partial derivatives ([59, 47]) 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 h one can show that $\operatorname{rank} M_h \geq r$. In particular, by basic linear algebra this shows that there is some $r \times r$ submatrix M'_h of M_h such that $\det M'_h \neq 0$, 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 h 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 [86].

1. *Usefulness:* For low-complexity f we have that $f \in P$ as argued above. Further, P is a non-trivial property as $h \notin P$.

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 ([17, 68]). Thus, the property P is efficiently decidable in the size of its input.

[2] Grochow [44] referred to distinguishers as *test polynomials*, as they test whether an input polynomial is of low- or high-complexity.

3. *Largeness:* The largeness condition (for the complement of the property P) 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 $\det M'_h \neq 0$. As non-zero polynomials evaluate to non-zero at random points with high probability ([94, 107, 28]), 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. One might expect the Razborov and Rudich [86] barrier to then rule out such properties, but their barrier result only holds when the underlying circuit class can compute pseudorandom functions. While it is widely believed 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. Hence, the Razborov and Rudich [86] 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 a 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. $\mathbb{F}[x_1,\ldots,x_n]^d_{\mathrm{hom}}$: The space of homogeneous n-variate polynomials of total degree exactly d. There are
$$N^{\mathrm{hom}}_{n,d} := \binom{n+d-1}{d}$$
many monomials $\mathbf{x}^{\mathbf{a}}$ in this space.

3. $\mathbb{F}[x_1,\ldots,x_n]^d_{\mathrm{ideg}}$: The space of n-variate polynomials of individual degree at most d. There are
$$N^{\mathrm{ideg}}_{n,d} := (d+1)^n$$
many monomials $\mathbf{x}^{\mathbf{a}}$ in this space.

While this may seem pedantic, it is important to distinguish these spaces. Even though homogeneous degree-d polynomials capture nearly all of the interesting complexity of polynomials of degree *at most* d, it is trivial to distinguish the two. For example, consider the distinguisher polynomial $c_{\mathbf{0}}$ that simply returns the constant coefficient (the coefficient of 1) of a polynomial $f = \sum_{\mathbf{a}} c_{\mathbf{a}}\mathbf{x}^{\mathbf{a}}$. This polynomial vanishes on $\mathbb{F}[x_1,\ldots,x_n]^d_{\mathrm{hom}}$ for $d > 0$, but does not vanish on the constant polynomial $1 \in \mathbb{F}[x_1,\ldots,x_n]^d$. However, it would be absurd to say that "1 is a hard polynomial for $\mathbb{F}[x_1,\ldots,x_n]^d_{\mathrm{hom}}$." Thus, in discussing how properties can distinguish polynomials we must specify the domain of interest. Indeed, to discuss lower bounds for homogeneous computation one must restrict attention to the space $\mathbb{F}[\mathbf{x}]^d_{\text{hom}}$, and likewise to discuss lower bounds for multilinear computation one must restrict attention to the space $\mathbb{F}[\mathbf{x}]^1_{\text{ideg}}$.

We now present our definition, with enough generality to handle the above spaces of polynomials simultaneously. 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 $\mathcal{M} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ 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 $\mathcal{C} \subseteq \text{span}(\mathcal{M})$ 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** $\mathcal{C}$, also called a **distinguisher**, if

1. D is a non-zero polynomial and
2. for all $f \in \mathcal{C}$, 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 $\mathcal{M}$ 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 "**coeff**" if the space of polynomials is clear from the context.

Thus, to revisit the comparison with Razborov and Rudich [86], condition (2) says that the distinguisher D is *useful* against the class $\mathcal{C}$. 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 ([94, 107, 28]). 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.

Further, note how the above distinguishers naturally have a *one-sided* nature to them as in algebraic complexity one typically seeks lower bounds against computations using *any* field extension of the base field of coefficients. In using the above to define the Razborov and Rudich [86] style property $P := \{f : D(\mathbf{coeff}_\mathcal{M}(f)) = 0\}$, we note that the complement property $\neg P = \text{span}(\mathcal{M}) \setminus P = \{f : D(\mathbf{coeff}_\mathcal{M}(f)) \neq 0\}$ *cannot* be expressed in the above framework: for non-zero polynomials p and q, it cannot be that the product pq vanishes everywhere (over large enough fields), so that in particular it cannot be that $p(\alpha) = 0$ iff $q(\alpha) \neq 0$.

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 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 [59, 47] is not currently known to be VP-natural. 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* avoid the algebraic framework (constructive or not). 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 ([23, Chapter 6], [18]) 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. 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 [94, 107, 28]. This lemma implies that 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 α and observe whether $D(\alpha) = 0$, 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 [55, 4, 56, 30].

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 $\alpha \in \mathcal{H}$ such that $D(\alpha) \neq 0$. 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. For more, see the surveys of Saxena [89, 90] and Shpilka-Yehudayoff [97].

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

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

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. In other words, 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 [105] in the Boolean setting, where the existence of the Razborov and Rudich [86] 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

$$\operatorname{span}(\mathcal{M}) := \left\{ \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F} \right\}$$

be all linear combinations of these monomials. Let $\mathcal{C} \subseteq \operatorname{span}(\mathcal{M})$ 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.

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

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

$$\operatorname{span}(\mathcal{M}) := \left\{ \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F} \right\}$$

be all linear combinations of these monomials. Let $\mathcal{C} \subseteq \operatorname{span}(\mathcal{M})$ 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 $\mathcal{C}$ iff $\mathcal{C}$ is not a $\mathcal{C}$-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 $\mathcal{C} \subseteq \mathbb{F}[x_1, \dots, x_n]^d$ be the class of $\operatorname{poly}(n,d)$-size circuits of total degree at most d. Then there is an algebraic $\operatorname{poly}(N_{n,d})$-natural proof against $\mathcal{C}$ iff $\mathcal{C}$ is not a $\operatorname{poly}(n,d)$-succinct hitting set for $\operatorname{poly}(N_{n,d})$-size circuits in $N_{n,d}$ variables.*

In the common regime when $d = \operatorname{poly}(n)$, we have that $\operatorname{poly}(n) = \operatorname{polylog}(N_{n,d})$ so that the existence of an algebraic natural proofs barrier is equivalent to saying that coefficient-vectors of polylogarithmic-size circuits (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 $\operatorname{polylog}(N)$-succinct hitting set for circuits of $\operatorname{poly}(N)$-size?

Again, we note that Question 1.6 was also raised by Grochow, Kumar, Saks, and Saraf [45], who presented a definition similar to Definition 1.3 and also observed the implication in Theorem 1.4.

Succinct generators. 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 $\mathcal{D} \subseteq \mathbb{F}[c_1, \dots, c_N]$ is to obtain a *generator*, which is a polynomial map $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^N$ such that $D \in \mathcal{D}$ is a non-zero iff $D \circ \mathcal{G} \not\equiv 0$, that is, the composition $D(\mathcal{G}(\mathbf{y})) \neq 0$ 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 [97].

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 $\operatorname{poly}(|\mathcal{H}|)$. However, as we are trying to hit polynomials on N variables, this would yield a $\operatorname{poly}(N)$-size generator whereas we would want a generator of complexity $\operatorname{polylog}(N)$. We now define succinct generators and give a tighter relationship with succinct hitting sets.

Definition 1.7. Let $\mathcal{M} \subseteq \mathbb{F}[x_1, \dots, x_n]$ be a set of monomials $\mathcal{M} = \{\mathbf{x}^{\mathbf{a}}\}_{\mathbf{a}}$, and let the set

$$\operatorname{span}(\mathcal{M}) := \left\{ \sum_{\mathbf{x}^{\mathbf{a}} \in \mathcal{M}} c_{\mathbf{a}} \mathbf{x}^{\mathbf{a}} : c_{\mathbf{a}} \in \mathbb{F} \right\}$$

be all linear combinations of these monomials. Let $\mathcal{C} \subseteq \operatorname{span}(\mathcal{M})$ 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 $\mathcal{C}' \subseteq \mathbb{F}[x_1, \dots, x_n, y_1, \dots, y_\ell]$ be another class of polynomials.

We say that a polynomial map $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^{\mathcal{M}}$ is a $\mathcal{C}$**-succinct generator for** $\mathcal{D}$ **computable in** $\mathcal{C}'$ 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 in $\mathcal{C}'$, 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 $\alpha\in\mathbb{F}^\ell$, the polynomial $G(\mathbf{x},\alpha)\in\mathcal{C}$.

3. $\mathcal{G}$ is a generator 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 $D(\mathbf{coeff}_\mathcal{M}(G(\mathbf{x},\mathbf{y})))\neq 0$ 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
$$\mathbf{coeff}_\mathcal{M}(G(\mathbf{x},\mathbf{y}))\in\mathbb{F}[\mathbf{y}]^\mathcal{M}.$$

Conditions (2) and (3) are equivalent, over large enough fields, to the property that the output of the generator $\mathcal{G}(\mathbf{x},\mathbb{F}^\ell)=\{G(\mathbf{x},\alpha):\alpha\in\mathbb{F}^\ell\}$ is a $\mathcal{C}$-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 $\mathcal{C}'$.

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

This definition bears a slight resemblance to Mulmuley's [70] definition of an "explicit variety." A discussion about the connections between our work and Geometric Complexity Theory appears in Section 1.7.

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* $\operatorname{polylog}(N)$*-size circuit which is a* $\mathsf{VP}_{\operatorname{polylog}(N)}$*-succinct generator for* VP_N *iff there is a* $\mathsf{VP}_{\operatorname{polylog}(N)}$*-succinct hitting set for* VP_N*. Further, the existence of any* $\mathsf{VP}_{\operatorname{polylog}(N)}$*-succinct hitting set for* VP_N *implies an explicit* $\operatorname{poly}(N)^{\operatorname{polylog}(N)}$*-size hitting set for* VP_N*.*

Note that Aaronson and Drucker [2] proposed a candidate algebraic pseudorandom function based on generic projections of determinants. Their construction does not seem sufficient for the above result, as discussed in Section 3.

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 ([14]) and algebraization ([3]), which are unconditional). Our belief in such barriers therefore rests on the plausibility of these conditional assumptions. We now review two sources of evidence, cryptographic and complexity-theoretic.

The Razborov and Rudich [86] paper showed that there is a natural proofs barrier under the assumption of the existence of pseudorandom functions with exponential security. As discussed in the Introduction, there are two good reasons to believe the plausibility of this assumption. First, is that there are many well-studied candidate constructions which are believed to have this security. Second, is that there is a web of security-preserving reductions between cryptographic notions, in particular showing that such pseudorandom functions follow from pseudorandom generators with exponential security ([43]) or even one-way functions with exponential security ([54]). One-way functions are the most basic cryptographic object, so that essentially the natural proofs barrier holds unless cryptography fails.[3]

The above cryptographic evidence already seems strong enough, but it is worth mentioning another evidence based on complexity-theoretic derandomization. For many classes of restricted computation there have been pseudorandom generators $\mathcal{G} : \{0,1\}^{\ell} \to \{0,1\}^n$ that fool these restricted classes even when $\ell = \mathrm{polylog}(n)$. For example, AC^0 is fooled by the Nisan-Wigderson [76] generator instantiated with parity ([75]) as well as by $\mathrm{polylog}(n)$-wise independence ([20]), RL is fooled by Nisan's [69] generator, and ε-bias spaces fool linear polynomials over $\mathbb{F}_2$ ([72, 12]). In each of these cases it turns out that the generators are in fact pseudorandom *functions* that fool these restricted classes, in that for every seed ℓ, $\mathcal{G}(\ell)$ can be thought of as a truth table of a function $\mathcal{G}_\ell : \{0,1\}^{\log n} \to \{0,1\}$, such that $\mathcal{G}_\ell$ can actually be computed in $\mathrm{poly}(\ell, \log n) = \mathrm{polylog}(n)$-time (as $\ell = \mathrm{polylog}(n)$). That is, these derandomization results actually provide succinct derandomization in the sense of Williams [105]. In fact, Razborov and Rudich [86] explicitly noted how Nisan's [75] pseudorandom generator for AC^0 is a pseudorandom function (with an application to how the lower bounds for $\mathsf{AC}^0[2]$ are thus provably more complicated than those for AC^0). It can be seen that fooling a restricted class of computation $\mathcal{C}$ using the Nisan-Wigderson [76] generator, when the hard function f against $\mathcal{C}$ is actually efficiently computable in P, gives rise to a pseudorandom function *unconditionally* secure against $\mathcal{C}$. In this case, each output of the Nisan-Wigderson generator can be thought of as a truth table of a simple function; this follows from the assumption on f and the fact that designs are efficiently computable, and see [24] for further discussion.

Algebraic complexity. Having reviewed the evidence for a natural proofs barrier in the Boolean setting, we can then ask: what evidence is there for an algebraic natural proofs barrier? Unfortunately, such evidence 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 [43] that converts a pseudorandom generator to a pseudorandom function seems to have no algebraic analogue ([2]) as this construction applied to

[3] Furthermore, there exist problems, such as the discrete logarithm problem, for which the natural proof barrier holds unconditionally.

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 paper 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 ([56]) 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 we posit the following meta-conjecture.

Meta-Conjecture 1.9. For any restricted 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 paper 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 defer the exact definitions of these models to the relevant sections. 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 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 following forms:*

- $\Sigma^{O(1)}\Pi\Sigma$ *formulas (Section 4.1),*
- $\Sigma\Pi\Sigma$ *formulas of transcendence degree* $\leq O(1)$ *(Section 4.2),*
- *sparse polynomials (Section 5.1),*
- $\Sigma\mathrm{m}\wedge\Sigma\Pi^{O(1)}$*-formulas (Section 5.2),*
- *commutative roABPs (Section 5.3),*
- *depth-*$O(1)$ *Occur-*$O(1)$ *formulas (Section 5.4),*
- *arbitrary circuits composed with sparse polynomials of transcendence degree* $O(1)$ *(Section 6).*

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, so 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 7)**.** *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.*

As commented above, while the *length* of the roABPs whose coefficient vectors define the hitting set is merely $n = \log(N)$, the *width* is as large as w^2, while a truly succinct hitting set would require the width to be $\mathrm{polylog}(w)$.

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 [63], which is often used in hitting-set constructions for more sophisticated algebraic computation. However, as we explain in Section 8, 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. So, in order 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 [63] 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 [96] 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 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 [42] 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.

We now briefly illustrate the simplest example of how we take existing constructions and make them succinct. Suppose one wanted to hit a non-zero linear polynomial $D(\mathbf{c}) = \alpha_1 c_1 + \cdots + \alpha_N c_N$. A standard approach would be to replace $c_i \leftarrow z^i$ where z is a new variable, as one now obtains a univariate polynomial $D(z) = \alpha_1 z^1 + \cdots + \alpha_N z^N$ which is clearly still non-zero. Now, however, there is simply one variable of degree N so that interpolation over this variable yields a hitting set of size $N+1$, which is essentially optimal in terms of hitting set size. To see how to make this succinct, note that the resulting vectors in the hitting set have the form $(\beta, \beta^2, \ldots, \beta^N)$ for $\beta \in \mathbb{F}$. For $N = 2^n$ so that we can identify $\mathbb{F}^N$ as the coefficient vectors of multilinear polynomials $\mathbb{F}[x_0, \ldots, x_{n-1}]^1_{\text{ideg}}$, we can see that such vectors can be succinctly represented as the coefficients of

$$\beta(1 + x_0 \beta^{2^0})(1 + x_1 \beta^{2^1}) \cdot (1 + x_{n-1} \beta^{2^{n-1}}) \, ,$$

using the fact that we can express each number in $\{1, \ldots, N\}$ uniquely in its binary representation. Further, we can even make this construction low-degree in all of the variables by considering

$$\beta(1 + x_0 \beta_0)(1 + x_1 \beta_1) \cdot (1 + x_{n-1} \beta_{n-1})$$

for new variables $\beta_0, \ldots, \beta_{n-1}$. This clearly embeds the previous construction so is still a hitting set, but is now the desired VP-succinct generator.

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 [71]. 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 $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]^d$, 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 $\mathcal{C}$, that is $\mathcal{D} = \{D : D(\mathbf{coeff}(f)) = 0, \ \forall f \in \mathcal{C}\}$. Thus, $\mathcal{D}$ vanishes on $\mathcal{C}$, but it also may vanish on other points. The *(Zariski) closure* of $\mathcal{C}$, denoted $\overline{\mathcal{C}}$, is simply all polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]^d$ which the distinguishers $\mathcal{D}$ vanish on, that is $\overline{\mathcal{C}} = \{f \in \mathbb{F}[x_1, \ldots, x_n]^d : D(\mathbf{coeff}(f)) = 0, \ \forall D \in \mathcal{D}\}$. Clearly $\mathcal{C} \subseteq \overline{\mathcal{C}}$, 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 $\mathbb{F}^2$ (for further examples related to algebraic complexity classes, see [21]).

From the perspective of algebraic geometry, it is much more natural to study the closure $\overline{\mathcal{C}}$ rather than the class $\mathcal{C}$ itself. And indeed, the algebraic natural proofs we define here necessarily give lower bounds for the closure $\overline{\mathcal{C}}$ because the lower bound is proven using a distinguisher in $\mathcal{D}$. In fact, algebraic geometry shows that lower bounds for $\overline{\mathcal{C}}$ *necessarily* must use such distinguishers (though they may not have small circuit size).[4] Thus, we see that 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. 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 $\mathcal{C}$ being computed but also their distinguishers $\mathcal{D}$. One can thus then ask that the lower bounds methods respect these symmetries, and Grochow [44] showed that most lower bounds in the literature do obey the natural symmetries one would expect. Although this is not exactly precise, a useful picture is that 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:

Corollary 1.12. *Let $\mathcal{T}$ be the set of defining equations for $\overline{\mathsf{VP}}$. For each of the models mentioned in Theorem 1.10, there exists a polynomial $P \in \mathcal{T}$ which requires super-polynomial size when computed in this model.*

1.8 Follow-up work

We end this section by briefly mentioning two related works that have appeared since the initial version of this paper was posted.

Efremenko, Garg, Oliveira and Wigderson [31] studied algebraic circuit lower bounds proved using subadditive complexity measures based on matrix rank. Such *rank-based* methods are often used in practice to prove lower bounds on restricted models of algebraic computation. These lower bounds are algebraic, and also often fall in our framework of algebraic natural proofs (as often the corresponding matrices are polynomially large in the relevant parameters, but this is not always true as seen in [47]). The main results of [31] are *unconditional* barriers on proving tensor-rank lower bounds or Waring-rank lower bounds using rank-based methods.

Bläser, Ikenmeyer, Jindal and Lysikov [19] studied our notion of algebraic natural proofs in the context of a complexity measure they call *border completion rank* of a affine linear matrix polynomial. They establish (among other results) that there is an infinite family of linear matrices for which no algebraically natural proof can prove the matrices have high border completion rank, assuming that the polynomial hierarchy does not collapse. This underlying assumption is more widely believed than the

[4] 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 [46]. It is conceivable that the algebraic distinguisher approach tries to prove too much, that is, perhaps $\overline{\mathsf{VP}} = \mathsf{VNP}$. We refer again to [21] for further discussion and examples which separate natural algebraic complexity classes from their closures.

conjecture that succinct hitting sets exist, but their conclusion does not rule out an algebraic natural proof for high border completion rank some some *other* set of matrices.

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 α,β are used to denote vectors of scalars. Similar boldface letters are used to denote tuples of polynomials. 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 $N^{\mathrm{ideg}}_{n,1}=2^n$ 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 $\mathbf{y}$.

3 Universal constructions of pseudorandom polynomials

In this section we detail *universal circuits* and their applications to pseudorandom polynomials. 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 α such that $f(\mathbf{x})=U(\mathbf{x},\alpha)$. 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 ([102]) (and hence complete for VP under quasipolynomial-size reductions ([103])). One would then expect that *if* there are pseudorandom polynomials then such universal circuits would also be pseudorandom.

Indeed, based on this intuition Aaronson and Drucker [2] gave a candidate construction of pseudorandom polynomials based on generic projections of the determinant, with the intention of exploiting the completeness of the determinant. However, we note here that while it is plausible that this construction is in fact pseudorandom, it is insufficient for our requirement for universality, as we want the computed f and the universal U to live in the same space of polynomials: if f is of low total-degree so that $f\in\mathbb{F}[x_1,\ldots,x_n]^d$, then we want $U(\mathbf{x},\alpha)\in\mathbb{F}[x_1,\ldots,x_n]^d$ for *every* α. This is because we want a collection of polynomials $\mathcal{C}$ that is indistinguishable from generic polynomials in $\mathbb{F}[x_1,\ldots,x_n]^d$. If we start with such a collection and attempt to embed them into U where $\deg_{\mathbf{x}} U(\mathbf{x},\mathbf{y})=d'\gg d$, the resulting collection of polynomials necessarily lives in the larger space $\mathbb{F}[\mathbf{x}]^{d'}$ and the indistinguishability property no longer clearly holds. As a concrete example, suppose $f(\mathbf{x})$ is a "generic" polynomial in $\mathbb{F}[\mathbf{x}]^d$. Then the modified polynomial $f(\mathbf{x})+z$ still embeds f, yet it lives in $\mathbb{F}[\mathbf{x},z]^d$, where it is no longer generic as it is linear in z.

Thus, we need a universal circuit construction that does not increase the degree of $\mathbf{x}$. For algebraic branching programs, the candidate of Aaronson and Drucker [2] is easy to fix by switching from the determinant to iterated matrix multiplication, which is also complete but due to efficient homogenization of branching programs ([74]) can be universal without increasing degree. However, for full generality we want to be universal for circuits, i. e., to obtain a polynomial U complete for VP under polynomial-size reductions which also ensures the $\mathbf{x}$-degree of U matches that of f. The first techniques for constructing universal circuits were given by Bürgisser [22, Section 5.6]. Later, Raz [81] developed slightly different techniques, which among other things were also able to respect circuit depth. We state a version below which is implicit in both works.

Theorem 3.1 (Bürgisser [22, Section 5.6], Raz [81])**.** *Let $\mathbb{F}$ be a field, and let $n,s \geq 1$ and $d \geq 0$. Then there is a $\mathrm{poly}(n,d,s)$-size algebraic circuit $U \in \mathbb{F}[x_1,\ldots,x_n,y_1,\ldots,y_r]$ with $r \leq \mathrm{poly}(n,d,s)$ such that U can be constructed in time $\mathrm{poly}(n,d,s)$, and*

- $\deg_{\mathbf{x}} U(\mathbf{x},\mathbf{y}) \leq d$,
- $\deg_{\mathbf{y}} U(\mathbf{x},\mathbf{y}) \leq \mathrm{poly}(d)$, *and*
- *if $f \in \mathbb{F}[\mathbf{x}]$ has $\deg_{\mathbf{x}} f \leq d$ and f is computed by a size s circuit, then there is some $\alpha \in \mathbb{F}^r$ such that $f(\mathbf{x}) = U(\mathbf{x},\alpha)$.*

We briefly note that this construction also yields a universal circuit for homogeneous degree-d computations (the space $\mathbb{F}[x_1,\ldots,x_n]^d_{\mathrm{hom}}$). No such universal circuits are known for efficient multilinear computation (the space $\mathbb{F}[x_1,\ldots,x_n]^1_{\mathrm{ideg}}$), 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 $\mathrm{poly}(n,d,s)$-succinct generator for $\mathcal{D}$ computable by $\mathrm{poly}(n,d,s)$-size circuits.*

Proof. Let the s-succinct hitting set arise from the set of size-s polynomials $\mathcal{C} \subseteq \mathbb{F}[x_1,\ldots,x_n]^d$. Let $U \in \mathbb{F}[\mathbf{x},y_1,\ldots,y_r]$ be the universal circuit of Theorem 3.1. Then for any $f \in \mathcal{C}$ there is some $\alpha \in \mathbb{F}^r$ such that $f(\mathbf{x}) = U(\mathbf{x},\alpha)$. Thus,

$$\mathcal{C} \subseteq U(\mathbf{x},\mathbb{F}^r) = \{U(\mathbf{x},\alpha) : \alpha \in \mathbb{F}^r\}\ .$$

Thus, we see that U is indeed a generator for $\mathcal{D}$ as it contains the hitting set $\mathcal{C}$ in its image. Further, $U(\mathbf{x},\alpha) \in \mathbb{F}[x_1,\ldots,x_n]^d$ for all $\alpha \in \mathbb{F}^r$ by construction. Finally $U(\mathbf{x},\alpha)$ is computable in size $\mathrm{poly}(n,d,s)$ for all $\alpha \in \mathbb{F}^r$ as $U(\mathbf{x},\mathbf{y})$ has such a circuit and the substitution $\mathbf{y} \leftarrow \alpha$ does not increase circuit size. It follows that U is the desired succinct generator. $\square$

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 $(\delta\Delta+1)^\ell$.*

Proof. For any $D \in \mathcal{D}$, we see that D is non-zero iff $D(\mathbf{coeff}_{\mathbf{x}} G(\mathbf{x},\mathbf{y}))$ is non-zero as a polynomial in $\mathbf{y}$. In particular, $\deg_{\mathbf{y}} D(\mathbf{coeff}_{\mathbf{x}} G(\mathbf{x},\mathbf{y})) \le \deg D \cdot \deg_{\mathbf{y}} G \le \delta\Delta$. Thus, as the field is large enough we can find a set $S \subseteq \mathbb{F}$ with $|S| \ge \delta\Delta + 1$, so that by interpolation $D(\mathbf{coeff}_{\mathbf{x}} G(\mathbf{x},\mathbf{y}))$ is non-zero iff $D(\mathbf{coeff}_{\mathbf{x}} G(\mathbf{x},\alpha))$ is non-zero for every $\alpha \in S^\ell$. Thus, we see that $G(\mathbf{x}, S^\ell)$ is the desired succinct hitting set as each $G(\mathbf{x},\alpha)$ has a size-s circuit (as substitution does not increase circuit size) and S^ℓ has the correct size. $\square$

In the usual range of parameters we would have $\Delta = \mathrm{poly}(N)$ and $\delta = \mathrm{poly}(n,s)$. 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 $n \ge 1$. 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 $\mathrm{poly}(N)$-size $\mathrm{poly}(N)$-degree circuits in $\mathbb{F}[c_1,\ldots,c_N]$ have $\mathrm{poly}(n)$-succinct hitting sets from $\mathbb{F}[\mathbf{x}]^n$, then such circuits have an explicit $\mathrm{poly}(N)^{\mathrm{polylog} N}$-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 [42], in the context of studying deterministic extractors, studied how to pseudo-randomly map $\mathbb{F}^n$ to $\mathbb{F}^r$ preserving vector spaces of dimension r with high probability. In particular, they gave a $\mathrm{poly}(n)$-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 [36], and was called a *rank condenser* in later work ([35, 34]) 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 [42] 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 $n \ge r \ge 1$. Define the polynomial $P^{\mathrm{RC}}_{n,r}$ where $P^{\mathrm{RC}}_{n,r} \in \mathbb{F}[x_1,\ldots,x_n,y_1,\ldots,y_r,t_0,t_1,\ldots,t_n]$ to be

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

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 [42].

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

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

Proof. We can index the output coordinates of $\mathcal{G}^{\mathrm{RC}}_{n,r}$ with subsets $S \subseteq [n]$, so that an index $i \in [N]$ gets mapped to $S \subseteq [n]$ via its binary representation so that $i-1 = \sum_{k\in S} 2^{k-1}$, and for a given $S \subseteq [n]$ denote the corresponding index i_S. Then,

$$\begin{aligned}
P^{\mathrm{RC}}_{n,r}(\mathbf{x},\mathbf{y},t,t^{2^0},t^{2^1}\cdots,t^{2^{n-1}}) &= \sum_{j=1}^{r} y_j t^j \prod_{k=1}^{n}(1+x_k(t^{2^{k-1}})^j) \\
&= \sum_{j=1}^{r} y_j t^j \sum_{S\subseteq[n]} \prod_{k\in S} x_k \cdot t^{j\cdot 2^{k-1}} \\
&= \sum_{j=1}^{r} y_j t^j \sum_{S\subseteq[n]} t^{j\cdot\sum_{k\in S}2^{k-1}} \prod_{k\in S} x_k \\
&= \sum_{j=1}^{r} y_j t^j \sum_{S\subseteq[n]} t^{j\cdot(i_S-1)} \prod_{k\in S} x_k .
\end{aligned}$$

Thus, taking coefficients in $\mathbf{x}$ exactly indexes $\sum_{j=1}^{r} y_j t^{ij}$ as required. □

We now observe that this generator is efficiently computable, and produces succinct hitting sets.

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

4.1 Depth-3 formulas with bounded top-fan-in

A $\Sigma^k\Pi\Sigma$ *formula* is a depth-3 formula of the form $\sum_{i=1}^{k}\prod_{j=1}^{d_i}\ell_{i,j}$, where $\ell_{i,j}$ are linear functions in $x_1,\ldots,x_N$. We denote the degree of the circuit by $d = \max_i d_i$.

The study of $\Sigma^k\Pi\Sigma$ formulas was initiated by Dvir and Shpilka [29], who proved that in a *simple* and *minimal*[5] $\Sigma^k\Pi\Sigma$ circuit computing the zero polynomial, the rank of the linear functions $\{\ell_{i,j}\}$ is bounded by a number $R(k,d)$ that is independent of the number of variables N. The number $R(k,d)$ is called the *rank bound* for this class of circuits. Karnin and Shpilka [57] showed how to use the rank condenser construction of Gabizon and Raz in order to obtain a black-box identity testing algorithm, and improved rank bounds were later obtained ([62, 91, 92, 93]).

In this section, we construct a $\mathrm{poly}(n,k)$-$\Sigma\Pi\Sigma$ succinct hitting set for $\Sigma^k\Pi\Sigma$ formulas, and we use the fact that the rank condenser generator, with a judicious choice of r, is a generator for $\Sigma^k\Pi\Sigma$ formulas. The version we cite here is from the survey [97].

[5] We omit the exact definitions here and refer the reader to [97] for a thorough discussion.

Fact 4.4 (Hitting set for $\Sigma^k\Pi\Sigma$ Formulas). *Let $F(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be a polynomial computed by a $\Sigma^k\Pi\Sigma$ degree d formula. Let $V : \mathbb{F}^r \to \mathbb{F}^N$ the linear transformation given by the $N \times r$ Vandermonde matrix $(V_t)_{ij} = t^{i \cdot j}$ for $1 \leq i \leq N$, $1 \leq j \leq r$. Then, for $r = R(k,d)+1$ where $R(k,d) = O(k^2 \log d)$ (over finite fields) or $R(k,d) = k^2$ (over infinite fields), it holds that $F \neq 0$ if and only if the r-variate polynomial $F \circ \left(V_t \cdot (y_1,\dots,y_r)^T\right)$ is non-zero.*

Using Fact 4.4 and the properties of Construction 4.1, we obtain the following two lemmas.

Lemma 4.5. *The polynomial map $\mathcal{G}^{\mathrm{RC}}_{n,R(k,d)}(\mathbf{y},\mathbf{t})$ is $\mathrm{poly}(R(k,d),n)$-$\Sigma\Pi\Sigma$ succinct. In particular, the generator is $\mathrm{poly}(k,\log d,n)$-$\Sigma\Pi\Sigma$ succinct.*

Proof. The first statement is immediate from Proposition 4.3. The second statement follows using the rank bounds for $\Sigma^k\Pi\Sigma$ formulas stated in Fact 4.4. □

Lemma 4.6. *Let F be computed by a $\Sigma^k\Pi\Sigma$ formula. Then $F \circ \mathcal{G}^{\mathrm{RC}}_{n,R(k,d)} \not\equiv 0$.*

Proof. Immediate from Proposition 4.2 (making the appropriate substitution for $\mathbf{t}$) and Fact 4.4. □

Corollary 4.7. *$\mathcal{G}^{\mathrm{RC}}_{n,R(k,d)}(\mathbf{y},\mathbf{t})$ is a $\mathrm{poly}(k,\log d,n)$-$\Sigma\Pi\Sigma$ succinct generator for the class of $\Sigma^k\Pi\Sigma$ formulas.*

4.2 Depth-3 circuits of bounded transcendence degree

We now generalize the results of Section 4.1 to obtain a succinct hitting set for the larger class of circuits with *bounded transcendence degree*.

A set of polynomials $\{F_1,\dots,F_r\} \subseteq \mathbb{F}[\mathbf{X}]$ is called *algebraically independent* if for any non-zero polynomial $H \in \mathbb{F}[w_1,\dots,w_r]$, $H(F_1,\dots,F_r) \not\equiv 0$. Given a set of polynomials $\{F_1,\dots,F_\ell\}$, the *transcendence degree* of this set, denoted $\mathsf{trdeg}\{F_1,\dots,F_\ell\}$, is the size of a maximal algebraically independent subset of $\{F_1,\dots,F_\ell\}$.

Let $C(Y_1,\dots,Y_M)$ be a circuit of polynomial degree, and for $i \in [m]$, let $T_i = \prod_{j=1}^d L_{i,j}$, where $L_{i,j} \in \mathbb{F}[X_1,\dots,X_N]$ are linear functions. In [6], Agrawal et al. present a hitting set for polynomials of the form $F = C(T_1,\dots,T_M)$, where $\mathsf{trdeg}\{T_1,\dots,T_m\}$ is bounded by k (the size of the hitting set is exponential in k). In this section we present a succinct version of their generator.

Lemma 4.8 (Generator for circuits of transcendence degree k, [6], and see also the presentation in Chapter 4 of [87]). *Suppose $\mathbb{F}$ is a field such that $\mathsf{char}(\mathbb{F}) = 0$ or $\mathsf{char}(\mathbb{F}) \geq d^k$. Then the map $\Psi : \mathbb{F}[\mathbf{X}] \to \mathbb{F}[y_1,\dots,y_k,t,z_1,\dots,z_k,s]$, given by*

$$X_i \mapsto \sum_{j=1}^{k+1} z_j s^{ij} + \sum_{j=1}^{k} y_j t^{ij}$$

for every $i \in [N]$, is a generator for the class of polynomials $F \in \mathbb{F}[\mathbf{X}]$ expressible as $C(T_1,\dots,T_M)$, where the T_i's are products of linear functions and $\mathsf{trdeg}\{T_1,\dots,T_m\} \leq k$.

It remains to be noted that we can construct the map Ψ succinctly.

Theorem 4.9. *Suppose $\mathbb{F}$ is a field such that $\mathsf{char}(F) = 0$ or $\mathsf{char}(F) \geq d^k$. Then there exists a $\mathsf{poly}(k,n)$-$\Sigma\Pi\Sigma$ succinct generator for the class of polynomials that can be represented as $C(T_1,\dots,T_M)$ with C being a $\mathsf{poly}(N)$ degree circuit, each T_i is a product of d linear functions and $\mathsf{trdeg}\{T_1,\dots,T_M\} \leq k$.*

Proof. Observe that Ψ from Lemma 4.8 can be represented as $\mathbf{coeff_x}(P(\mathbf{y},\mathbf{z},\mathbf{s},\mathbf{t}))$, where

$$P(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{s},\mathbf{t}) = P^{\mathsf{RC}}_{n,k+1}(\mathbf{x},\mathbf{z},\mathbf{s}) + P^{\mathsf{RC}}_{n,k}(\mathbf{x},\mathbf{y},\mathbf{t})\ .$$

The succinctness follows from Proposition 4.3, and from observing that $\mathrm{poly}(k,n)$-$\Sigma\Pi\Sigma$ circuits are closed under addition. □

5 Succinct hitting sets via the Shpilka-Volkovich generator

The Shpilka-Volkovich Generator (SV Generator, henceforth, and see [96]) is a polynomial map $\mathcal{G}(y_1,\dots,y_k,z_1,\dots,z_k) : \mathbb{F}^{2k} \to \mathbb{F}^N$ that satisfies the property that for every $T \subseteq [N]$ such that $|T| \leq k$, we can set $z_1,\dots,z_k$ to values $\alpha_{i_1},\dots,\alpha_{i_k}$ 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

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

and

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

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

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

We begin by stating some immediate facts regarding Construction 5.1.

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

Fact 5.3 (Additivity)**.** *The succinct SV-generator is additive in $\mathbf{y},\mathbf{z}$, in the sense that as polynomials, we have the equality*

$$Q^{\mathsf{SSV}}_{n,k_1}(\mathbf{y}_1,\mathbf{z}_1,\mathbf{x}) + Q^{\mathsf{SSV}}_{n,k_2}(\mathbf{y}_2,\mathbf{z}_2,\mathbf{x}) = Q^{\mathsf{SSV}}_{n,k_1+k_2}(\mathbf{y}',\mathbf{z}',\mathbf{x})\ ,$$

where $\mathbf{y}' = (\mathbf{y}_1,\mathbf{y}_2)$ and $\mathbf{z}' = (\mathbf{z}_1,\mathbf{z}_2)$. In particular, since the mapping from a polynomial to the coefficients vector is linear, as polynomial maps we get the equality

$$\mathcal{G}^{\mathsf{SSV}}_{n,k_1}(\mathbf{y}_1,\mathbf{z}_1) + \mathcal{G}^{\mathsf{SSV}}_{n,k_2}(\mathbf{y}_2,\mathbf{z}_2) = \mathcal{G}^{\mathsf{SSV}}_{n,k_1+k_2}(\mathbf{y}',\mathbf{z}')\ .$$

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

Lemma 5.4. *For every $T \subseteq [N]$ such that $|T| \le k$, 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.*

Proof. As before, it is convenient to think of a subset of the N coordinates as family of subsets of $[n]$.

Since $\mathcal{G}^{\mathrm{SSV}}_{n,k}$ is given by the coefficients map of the polynomial $Q^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z},\mathbf{x})$, an equivalent form of interpreting the statement of the lemma is that we want to fix the $\mathbf{z}$ variables such that distinct $\mathbf{y}$ variables become the coefficients of the monomials $\mathbf{x}_S$, for $S \in T$, and the coefficients of all monomials not in T are zero.

Suppose first $|T| = k$ and denote $T = \{S_1,\dots,S_k\}$. For every $j \in [k]$ set $\mathbf{z}_j = \mathbf{1}_{S_j}$, the characteristic vector of the set $S_j \subseteq [n]$. That is, $z_{j,i} = 1$ if $i \in S_j$, and 0 otherwise.

Observe that, in the notation of Construction 5.1, we have that

$$P(\mathbf{1}_{S_j}, x_1,\dots,x_n) = \prod_{i=1}^{n}((\mathbf{1}_{S_j})_i \cdot x_i + (1-(\mathbf{1}_{S_j})_i)) = \prod_{i:(\mathbf{1}_{S_j})_i=1} x_i = \prod_{i\in S_j} x_i = \mathbf{x}_{S_j} \,.$$

Therefore, we get that

$$Q^{\mathrm{SSV}}_{n,k}(y_1,\dots,y_k,\mathbf{1}_{S_1},\dots,\mathbf{1}_{S_k},\mathbf{x}) = \sum_{i\in[k]} y_i \mathbf{x}_{S_i} \,,$$

as we wanted.

If $|T| = k' < k$, we can arbitrarily extend T so a set T' of size exactly k, and then set some $\mathbf{y}$ variables to zero, in order to zero out the relevant $k-k'$ entries in the polynomial map. □

Suppose we aim to hit a polynomial $F \in \mathbb{F}[\mathbf{X}]$ of degree d, and we are given the information that F contains a non-zero monomial with at most k variables. Assuming k is small, a natural algorithm in that case is to "guess" the $m \le k$ variables in the small support monomial, zero out all the remaining variables, and then do use the trivial derandomization, using the Schwartz-Zippel-DeMillo-Lipton Lemma, with respect to the remaining k-variate polynomial, for a cost of $(d+1)^k$ many evaluations. This is exactly what the SV generator enables us to do, since we can set the $\mathbf{z}$ variables in a way that the k $\mathbf{y}$ variables will contain those that appear in the small support monomial, and thus, since after fixing the $\mathbf{z}$ variables the polynomial remains non-zero, it follows that it is non-zero even without fixing the $\mathbf{z}$ variables. In this subsection we use this simple idea to construct succinct hitting sets for several classes of circuits. A small caveat is that usually we are *not* guaranteed our target polynomial has a small support monomial, but we can prove that this is the case after a proper shift of the N variables (one of course also has to represent the shift succinctly in n).

A similar notion was used by Agrawal, Saha and Saxena [7] to show that certain classes of polynomials simplify under shifts in a way which is helpful for designing PIT algorithms. In their case, the shift is by a vector of polynomials in a set of formal variables $\mathbf{t}$, whereas in our case the shifts are much simpler: for our applications we only need to shift by the constant vector $\mathbf{1}$.

Construction 5.5 (Succinct Hitting Set for classes with small support monomials after shifts by $\mathbf{1}$)**.** Let $k,n \in \mathbb{N}$ and $N = 2^n$. Define the *shifted succinct SV polynomial* to be

$$Q^{\mathrm{SSSV}}_{n,k}(y_1,\ldots,y_k,z_{1,1},\ldots,z_{1,n},\ldots,z_{k,1},\ldots,z_{k,n},x_1,\ldots,x_n) = Q^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z},\mathbf{x}) + \prod_{i=1}^{n}(x_i+1) \ ,$$

and the *shifted succinct SV generator* as

$$\mathcal{G}^{\mathrm{SSSV}}_{n,k}(\mathbf{y},\mathbf{z}) = \mathbf{coeff}_{\mathbf{x}}(Q^{\mathrm{SSSV}}_{n,k}) \ .$$

We record the following simple fact, which follows immediately from Fact 5.2, and from the fact that $\mathbf{coeff}(\prod_{i=1}^{n}(x_i+1)) = \mathbf{1}$.

Fact 5.6. *The generator $\mathcal{G}^{\mathrm{SSSV}}_{n,k}$ is $\mathrm{poly}(k,n)$-$\Sigma\Pi\Sigma$ succinct, and as polynomial maps, we have the equality*

$$\mathcal{G}^{\mathrm{SSSV}}_{n,k}(\mathbf{y},\mathbf{z}) = \mathcal{G}^{\mathrm{SSV}}_{n,k} + \mathbf{1} \ .$$

The following lemma shows how the shifted SV generator is useful for hitting classes of polynomials that have small support monomials after shifting by $\mathbf{1}$.

Lemma 5.7. *Let $\mathcal{C}$ be a class such that for all $f \in \mathcal{C}$, $F(\mathbf{X}+1)$ contains a monomial of support at most k. Then if $F \not\equiv 0$, $F \circ \mathcal{G}^{\mathrm{SSSV}}_{n,k}(\mathbf{y},\mathbf{z}) \not\equiv 0$.*

Proof. Let $F(\mathbf{X})$ be a non-zero polynomial from $\mathcal{C}$, and let $G(\mathbf{X}) = f(\mathbf{X}+\mathbf{1})$. By the assumption, G is a non-zero polynomial that contains a monomial M of support at most k. Let $S = \left\{X_{i_1},\ldots,X_{i_{k'}}\right\}$ (where possibly $k' < k$) denote the subset containing exactly the variables in M, and consider $G \circ \mathcal{G}^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z})$. By Lemma 5.4, we can set the $\mathbf{z}$ variables to α and possibly some of the $\mathbf{y}$ variables to β such that $y_1,\ldots,y_{k'}$ are mapped to $X_{i_1},\ldots,X_{i_{k'}}$, and all the other variables are mapped to 0. Under this setting

$$g \circ \mathcal{G}^{\mathrm{SSV}}_{n,k}(y_1,\ldots,y_{k'},\alpha,\beta) \not\equiv 0 \ ,$$

since the monomial M is mapped to a monomial in $y_1,\ldots,y_{k'}$ which cannot be canceled out. Hence, $G \circ \mathcal{G}^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z}) \not\equiv 0$.

Finally, observe that $F \circ \mathcal{G}^{\mathrm{SSSV}}_{n,k}(\mathbf{y},\mathbf{z}) = G \circ \mathcal{G}^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z})$. ☐

5.1 Sparse polynomials

In this section we give a $\mathrm{poly}(n)$-$\Sigma\Pi\Sigma$ succinct hitting set for the class of $\mathrm{poly}(N)$-sparse polynomials, i. e., polynomial size $\Sigma\Pi$ circuits. We note that an s-sparse polynomial f can also be computed by a commutative roABP of width s, so in a sense, the results in this section are subsumed by those in Section 5.3. However, the argument made here is simpler and slightly more general since it applies to any class that has (possibly after shifting) small support monomials (see Section 5.2).

We begin by recording the following fact.

Lemma 5.8 ([33, 50])**.** *Let $F \in \mathbb{F}[X_1,\ldots,X_N]$ by a polynomial with at most s monomials, and $\alpha \in \mathbb{F}^N$ be a full support vector, that is, for all $i \in [N]$, $\alpha_i \neq 0$. Then the polynomial $F(\mathbf{X}+\alpha)$ has a monomial of support at most $\log s$.*

This lemma appears in [33] and [50] with two very different proofs. For completeness, we provide yet a third proof, which we find to be more elementary. The proof relies upon the following easy lemma, due to Oliveira, which can be proved by induction on N (see, e. g., [38]).

Lemma 5.9 (see Proposition 6.14 in [38]). *Let $F \in \mathbb{F}[X_1,\dots,X_N]$ be a multilinear polynomial with at most s monomials, and $G \in \mathbb{F}[X_1,\dots,X_N]$ be any non-zero polynomial. Then $F \cdot G$ has at most s monomials.*

We now give our proof for Lemma 5.8.

Proof of Lemma 5.8. Suppose, towards contradiction, that the minimal monomial in $G(\mathbf{X}) := F(\mathbf{X}+\alpha)$ has $\ell \geq \log s + 1$ variables. Further suppose, without loss of generality, these are $X_1,\dots,X_\ell$. Consider now $G(X_1,X_2,\dots,X_\ell,0,\dots,0)$. By assumption, this is a non-zero polynomial which is divisible by the monomial $X_1X_2\cdots X_\ell$. It follows that

$$F(X_1,\dots,X_\ell,\alpha_{\ell+1},\dots,\alpha_n) = G(X_1-\alpha_1,\dots,X_\ell-\alpha_\ell,0,\dots,0) = \left(\prod_{i=1}^{\ell}(X_i-\alpha_i)\right)\cdot H(X_1,\dots,X_\ell)\,,$$

for some non-zero H.

Since $\prod_{i=1}^{\ell}(X_i-\alpha_i)$ is multilinear of sparsity $2^\ell > s$, it follows from Lemma 5.9 that the sparsity of $F(X_1,\dots,X_\ell,\alpha_{\ell+1},\dots,\alpha_n)$ is also greater than s, which contradicts the assumption on F, as the sparsity can only decrease when fixing variables. □

Lemma 5.8, along with Lemma 5.7 and Fact 5.6 immediately imply that the shifted succinct SV generator hits sparse polynomials.

Corollary 5.10. *The generator $\mathcal{G}^{\mathrm{SSSV}}_{n,\log s}$ from Construction 5.5 is a $\mathrm{poly}(\log s, n)$-$\Sigma\Pi\Sigma$ succinct generator for the class of s-sparse polynomials $F \in \mathbb{F}[X_1,\dots,X_N]$.*

5.2 Sums of powers of low degree polynomials

We now mention another class that, after a suitable shifting, has small support monomials.

Definition 5.11 ($\Sigma\mathrm{m}\wedge\Sigma\Pi^t$ formulas). A polynomial $F(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ is computed by a $\Sigma\mathrm{m}\wedge\Sigma\Pi^t$ formula if

$$F(\mathbf{X}) = \sum_{i=1}^{s} \mathbf{X}^{\mathbf{a}_i} F_i(\mathbf{X})^{d_i}\,,$$

where $\deg F_i \leq t$ for all $i \in [s]$, and $\mathbf{X}^{\mathbf{a}_i} = \prod_{j=1}^{N} X_i^{a_{i,j}}$ is a monomial.

The following was proved in [33].

Lemma 5.12. *Suppose $F[\mathbf{X}]$ is computed by a $\Sigma\mathrm{m}\wedge\Sigma\Pi^{O(1)}$ formula of top fan-in s, and let α be a full-support vector. Then it holds that $F(\mathbf{X}+\alpha)$ has a monomial of support at most $O(\log s)$.*

It follows that a similar construction to the one which we used to succinctly hit sparse polynomials also works in this case.

Theorem 5.13. *There exists a* $\mathrm{poly}(\log s, n)$*-*$\Sigma\Pi\Sigma$ *succinct generator for the class of* $\Sigma\mathrm{m}\wedge\Sigma\Pi^{O(1)}$ *formulas of top fan-in s.*

Proof. Let $C \cdot \log s$ the sparsity bound in Lemma 5.12 and consider the generator $\mathcal{G}^{\mathrm{SSSV}}_{n,C\log s}$ from Construction 5.5. By Fact 5.6, this generator is $\mathrm{poly}(\log s, n)$-$\Sigma\Pi\Sigma$ succinct. By Lemma 5.12 and Lemma 5.7, it follows that $\mathcal{G}^{\mathrm{SSSV}}_{n,C\log s}$ hits this class. □

5.3 Commutative read-once oblivious algebraic branching programs

In this section, we construct a $\mathrm{poly}(\log w, n)$-$\Sigma\Pi\Sigma$ succinct hitting sets for the class of N-variate polynomials computed by a width w *commutative* read-once oblivious algebraic branching programs.

A read-once oblivious algebraic branching program (roABP) is a directed, acyclic graph with the following properties:

- The vertices are partitioned into $N+1$ layers $V_0, \dots, V_N$, such that $V_0 = \{s\}$ and $V_N = \{t\}$. s is called the *source node*, and t the *sink node*.
- Each edge goes from V_{i-1} to V_i for some $i \in [N]$.
- There exists a permutation $\sigma : [N] \to [N]$ such that all edges in layer i are labeled by a univariate polynomial in $X_{\sigma(i)}$ of degree at most d.

We say that each $s \to t$ path in the ABP computes the product of its edge labels, and the roABP computes the sum over the polynomials computed by all $s \to t$ paths. The width of the roABP is defined to be $\max_i |V_i|$.

Equivalently, F is computed by a roABP in variable order σ if there are N matrices $M_1, \dots, M_N$ of size $r \times r$ such that each entry in M_i is a univariate, degree d polynomial in $X_{\sigma(i)}$, and

$$F = \left(\prod_{i=1}^{N} M_i(X_{\sigma(i)}) \right)_{1,1} .$$

In general, it is possible for a polynomial to be computed by a small width roABP in a certain variable order, but to require a much larger width if the roABP is in a different variable order. A polynomial $f \in \mathbb{F}[\mathbf{X}]$ is computed by a width w commutative roABP, if it is computable by a width w ABP in *every* variable order.

Forbes, Saptharishi and Shpilka ([35], Corollary 4.3) showed that in order to hit width-w commutative roABPs, it is enough to take the SV generator with $k = O(\log w)$.[6]

We follow the proof strategy of [35] in order to show that the succinct SV generator hits commutative roABPs as well. The following definitions and the theorem following them are borrowed from [35].

[6] An improved construction for this model, with respect to the size of the hitting set, was given by Gurjar, Korwar and Saxena [49]. Their construction, however, uses ingredients which we do not know how to make succinct; see Section 8 for further discussion.

Definition 5.14. Let $g : \mathbb{F}^m \times \mathbb{F}^{m'} \to \mathbb{F}^N$ be a polynomial map. g is said to be an *individual degree d, ℓ-wise independent monomial map* if for every $S \subseteq [N]$ of size at most ℓ, there is $\alpha \in \mathbb{F}^{m'}$ such that the polynomials $\{g(\mathbf{t},\alpha)^{\mathbf{a}} \,:\, \operatorname{supp}(\mathbf{a}) \subseteq S, \max_i a_i \le d\}$ are non-zero and distinct monomials in $\mathbf{t}$, where we define

$$g(\mathbf{t},\alpha)^{\mathbf{a}} = \prod_{i=1}^{N} (g(\mathbf{t},\alpha)_i^{a_i}) \; .$$

Definition 5.15 (see also [7])**.** Let $\mathbf{F}[\mathbf{X}] \in \mathbb{F}[\mathbf{X}]^r$ be a vector of polynomials. We say that $\mathbf{F}$ has support-k rank concentration at $\mathbf{v}$, if the derivatives of $\mathbf{F}$ with respect to all monomials of support at most k at $\mathbf{v}$ span all the derivatives of $\mathbf{F}$ at $\mathbf{v}$. That is, if

$$\operatorname{span}\{\partial_{\mathbf{X}^{\mathbf{a}}}(\mathbf{F})(\mathbf{v})\}_{\{\mathbf{a}:|\operatorname{supp}(\mathbf{a})|\le k\}} = \operatorname{span}\{\partial_{\mathbf{X}^{\mathbf{a}}}(\mathbf{F})(\mathbf{v})\}_{\mathbf{a}} \; .$$

Lemma 5.16 ([35], Theorem 4.1)**.** *Let $\mathbf{F}[\mathbf{X}] \in \mathbb{F}[\mathbf{X}]^{w\times w}$ be of individual degree d and computed by a commutative roABP of width w. Let $g(\mathbf{t},\mathbf{s})$ be an individual degree d, $(\log(w^2)+1)$-wise independent monomial map. Then $\mathbf{F}(\mathbf{X})$ has support-$\log(w^2)$ rank concentration at $g(\mathbf{t},\mathbf{s})$ over $\mathbb{F}(\mathbf{t},\mathbf{s})$.*

The succinct SV generator, like the SV generator, is a k-wise independent monomial map for and degree d.

Lemma 5.17. *The polynomial map $\mathcal{G}^{\mathrm{SSV}}_{n,k}(\mathbf{y},\mathbf{z})$ of Construction 5.1 is an individual degree d, k-wise independent monomial map for every d.*

Proof. By Lemma 5.4, for any set S of coordinates of size at most k we can set the $\mathbf{z}$ variables such that each coordinate in S contains a distinct y variable. Then, it is also clear that all individual degree up to d monomials of this map are distinct, for any choice of d □

The final ingredient (also from [35]) is the following lemma, which shows how to obtain hitting sets from rank concentration.

Lemma 5.18 ([35], Corollary 3.5)**.** *Let $\mathbf{F} \in \mathbb{F}[\mathbf{X}]^{r\times r}$ be a matrix of polynomials that is support-k rank concentrated at $\alpha \in \mathbb{F}^N$, and let $G(\mathbf{X}) = \mathbf{F}_{1,1}$. Then $G(\mathbf{X}) \not\equiv 0$ if and only if*

$$G \circ (\mathcal{G}^{\mathrm{SSV}}_{n,k} + \alpha) \not\equiv 0 \; .$$

We remark that although [35] phrase this lemma for their construction of the SV generator, the proof goes through verbatim using the properties of $\mathcal{G}^{\mathrm{SSV}}_{n,k}$ as explained in the proof of Lemma 5.17, and does not depend on the specific implementation.

We now prove that $\mathcal{G}_{n,4\log w+1}$ hits N-variate polynomials that are computed by width w commutative roABPs.

Theorem 5.19. *Let $|\mathbb{F}| > nd$, and let $F(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ be an N-variate polynomial of individual degree at most d, and computed by a width w commutative roABP. Then, $F \not\equiv 0$ if and only if*

$$F \circ \mathcal{G}^{\mathrm{SSV}}_{n,1+4\log w} \not\equiv 0 \; .$$

Proof. By definition, F is the $(1,1)$ entry of a matrix polynomial $\mathbf{F}(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]^{w\times w}$, with $\mathbf{F}$ being computed by a width-w commutative roABP. By Lemma 5.17 and Lemma 5.16, we get that the polynomial $\mathbf{F} \circ \mathcal{G}^{\mathrm{SSV}}_{n,2\log w+1}$ is support-$\log(w^2)$ rank concentrated. By Lemma 5.18, we deduce that $F \not\equiv 0$ if and only if

$$F \circ (\mathcal{G}^{\mathrm{SSV}}_{n,2\log w+1}(\mathbf{y}_1,\mathbf{z}_1) + \mathcal{G}^{\mathrm{SSV}}_{n,2\log w}(\mathbf{y}_2,\mathbf{z}_2)) \not\equiv 0 \ ,$$

where $\mathbf{y}_1,\mathbf{y}_2,\mathbf{z}_1,\mathbf{z}_2$ are disjoint sets of variables. By the additivity property (Fact 5.3), it holds that $F \not\equiv 0$ if and only if

$$F \circ \mathcal{G}^{\mathrm{SSV}}_{n,1+4\log w}(\mathbf{y},\mathbf{z}) \not\equiv 0$$

for $\mathbf{y} = (\mathbf{y}_1,\mathbf{y}_2)$ and $\mathbf{z} = (\mathbf{z}_1,\mathbf{z}_2)$. □

Corollary 5.20. *There exists a* $\mathrm{poly}(n,\log(w))$-$\Sigma\Pi\Sigma$ *succinct generator for the class of polynomials computed by width-w commutative roABPs.*

5.4 Depth-D occur-k formulas

The following model was considered in [6].

Definition 5.21. An *occur-k formula* is a directed tree, with internal nodes labeled either by $+$ or $\times\wedge$ (a power-product gate). The edges entering a $\times\wedge$ gate are labeled by integers $e_1,\dots,e_m$, and on inputs $g_1,\dots,g_m$, the gate computes $g_1^{e_1}\cdots g_m^{e_m}$. The leaves of tree are depth-2 formulas which compute sparse polynomials, such that every variable X_i occur in at most k of them.

The *size* of an occur-k formula is the sum over the sizes of its gates, where

1. the size of a "$+$" gate is 1,
2. the size of a "$\times\wedge$" gate is the sum $e_1+\cdots+e_m$ of the labels of its incoming edges, and
3. the size of a leaf node is the size of the depth-2 formula it is computing.

The *depth* of an occur-k formula is the number of layers of $+$ and $\times\wedge$ gates, plus 2, to account for the sparse formulas at the leaves.

Agrawal et al. ([6]) constructed a hitting set for this class, which combines both the rank condenser construction (Construction 4.1) and a generator for sparse polynomials. While the original construction uses the Klivans-Spielman generator ([63]), it is possible to make the hitting set succinct while using our version of the shifted succinct Shpilka-Volkovich generator.

We now present the succinct generator of depth-D occur-k formulas.

Construction 5.22. Let $D,k,n,s \in \mathbb{N}$. Denote $R = (2k)^{2D\cdot 2^D}$. For every $\ell \in [D-2]$, let $\mathbf{y}_\ell = (y_{\ell,1},\dots,y_{\ell,R})$ denote a tuple of R variables. We define the polynomial

$$P^{\mathrm{ASSS}}(\mathbf{x},\mathbf{y}_1,\dots,\mathbf{y}_{D-2},\mathbf{t}_1,\dots,\mathbf{t}_\ell,\mathbf{u},\mathbf{v}) = \sum_{\ell=1}^{D-2} \mathcal{G}^{\mathrm{RC}}_{n,R}(x_1,\dots,x_n,\mathbf{y}_\ell,\mathbf{t}_\ell) + Q^{\mathrm{SSSV}}_{n,R\log s+R\log R}(\mathbf{x},\mathbf{u},\mathbf{v}) \ ,$$

and the generator

$$\mathcal{G}^{\mathrm{ASSS}}(\mathbf{y}_1,\dots,\mathbf{y}_{D-2},\mathbf{t}_1,\dots,\mathbf{t}_\ell,\mathbf{u},\mathbf{v}) = \mathbf{coeff}_{\mathbf{x}}(P^{\mathrm{ASSS}}) \ .$$

In our setting, we think of $k, D = O(1)$, which immediately implies:

Fact 5.23. *For $k, D = O(1)$, the generator of Construction 5.22 is* $\mathrm{poly}(\log s, n)$-$\Sigma\Pi\Sigma$ *succinct.*

We now quote (a variant of) a theorem proved by Agrawal et al., which shows that Construction 5.22 is a generator for depth-D occur-k formulas.

Theorem 5.24 ([6], and see also the presentation in Chapter 4 of [87])**.** *Suppose $\Phi(\mathbf{w}) : \mathbb{F}^m \to \mathbb{F}^N$ is a map such that for any polynomial $F(\mathbf{X}) \in \mathbb{F}[\mathbf{X}]$ of sparsity at most $R! \cdot s^R$, $F \circ \Phi \not\equiv 0$. Then there exist integers $r_1, \dots, r_{D-2} \in [R]$, for $R = (2k)^{2D \cdot 2^D}$ such that the map*

$$\Psi : X_i \mapsto \sum_{\ell=1}^{D-2} \left(\sum_{j=1}^{r_\ell} y_{j,\ell} t_\ell^{ij} \right) + \Phi(\mathbf{w}) \tag{5.1}$$

is a generator for polynomials computed by depth-D occur-k formulas of size s assuming $\mathsf{char}(\mathbb{F}) = 0$ *or* $\mathsf{char}(\mathbb{F}) > s^R$.

As a corollary, we obtain that Construction 5.22 is a succinct generator for this class.

Corollary 5.25. *For $D, k = O(1)$, Construction 5.22 is a* $\mathrm{poly}(\log s, n)$-$\Sigma\Pi\Sigma$ *succinct generator for the class of polynomials computed by size-s depth-D occur-k formulas.*

Proof. The succinctness claim follows from Fact 5.23.

For the hitting property, observe that by Corollary 5.10, the polynomial map $\mathcal{G}^{\mathrm{SSSV}}_{n,m}(\mathbf{u}, \mathbf{v})$ satisfies the properties required from Φ in Theorem 5.24 for $m = R \log s + R \log R$, and by Proposition 4.2, the generator

$$\sum_{\ell=1}^{D-2} \mathcal{G}^{\mathrm{RC}}_{n,R}(\mathbf{y}_\ell, \mathbf{t}_\ell)$$

maps every X_i to the polynomial

$$\sum_{\ell=1}^{D-2} \left(\sum_{j=1}^{R} y_{j,l} t_\ell^{ij} \right)$$

after using the substitutions in $\mathbf{t}_\ell$ to a new variable t_ℓ as given in Proposition 4.2. Since $r_\ell \le R$, the polynomial in (5.1) is a projection of Construction 5.22, by possibly restricting excess $\mathbf{y}_\ell$ variables to 0.

The claim now follows from Theorem 5.24. $\square$

6 Succinct hitting sets for circuits of sparsely small transcendence degree

Another model, which was considered in [16], is that of circuits of the form $C(F_1, \dots, F_m)$ where the f_i's are polynomials of maximal sparsity s, $\mathsf{trdeg}\{F_1, \dots, F_m\} = r$ and C is an arbitrary circuit. It is possible to simplify the construction using ideas from [6], and thus we cite some of the definitions and the lemmas in the latter paper. Since we do not provide full proofs and do not discuss the full background, our terminology is slightly different at certain points.

We begin with the definition of the Jacobian matrix.

Definition 6.1. Let $\mathbf{F} = \{F_1(\mathbf{X}), \ldots, F_m(\mathbf{X})\} \subseteq \mathbb{F}[\mathbf{X}]$ be a set of N-variate polynomials. The Jacobian matrix of $\mathbf{F}$, denoted $\mathcal{J}_{\mathbf{X}}(\mathbf{F})$, is an $m \times N$ matrix such that $\mathcal{J}_{\mathbf{X}}(\mathbf{F})_{i,j} = \partial F_i / \partial X_j$.

The rank of the Jacobian matrix captures the transcendence degree of $\mathbf{F}$, assuming the characteristic is 0 or large enough.

Fact 6.2 ([16]). *Let $\mathbf{F} = \{F_1(\mathbf{X}), \ldots, F_m(\mathbf{X})\} \subseteq \mathbb{F}[\mathbf{X}]$ be a set of N-variate polynomials over $\mathbb{F}$ of degree at most d, such that $\mathsf{trdeg}\{F_1, \ldots, F_m\} = r$. If $\mathsf{char}(\mathbb{F}) = 0$ or $\mathsf{char}(\mathbb{F}) \geq d^r$, then $\mathrm{rank}_{\mathbb{F}(\mathbf{X})} \mathcal{J}_{\mathbf{X}}(\mathbf{F}) = r$.*

This fact shows that a map that preserves the rank of the Jacobian also preserves the transcendence degree of the f_i's, a fact which is useful for constructing generators (this is slightly non-trivial, and see [6] for details and discussion). For this purpose, we use the following "recipe" from [6] that gives a construction of such a map.

Lemma 6.3 ([6]). *Let $\mathbf{F} = \{F_1(\mathbf{X}), \ldots, F_m(\mathbf{X})\} \subseteq \mathbb{F}[\mathbf{X}]$ be a set of N-variate polynomials over $\mathbb{F}$ of degree at most d, such that $\mathsf{trdeg}\{F_1, \ldots, F_m\} \leq r$, and suppose $\mathsf{char}(\mathbb{F}) = 0$ or $\mathsf{char}(\mathbb{F}) \geq d^r$. Let $C(y_1, \ldots, y_m) \in \mathbb{F}[\mathbf{y}]$ be any polynomial, and let $\Phi : \mathbb{F}[\mathbf{X}] \to \mathbb{F}[\mathbf{z}]$ be a homomorphism such that*

$$\mathrm{rank}_{\mathbb{F}(\mathbf{X})}(\mathcal{J}_{\mathbf{X}}(\mathbf{F})) = \mathrm{rank}_{\mathbb{F}(\mathbf{z})} \Phi(\mathcal{J}_{\mathbf{X}}(\mathbf{F})) .$$

Consider the mapping Ψ given by

$$X_i \mapsto \left(\sum_{i=1}^{r} y_j t^{ij} \right) + \Phi(X_i) .$$

Then, it holds that $C(F_1, \ldots, F_m) \not\equiv 0$ if and only if $C(\Psi(F_1), \ldots, \Psi(F_m)) \not\equiv 0$.

We now show how to construct a succinct generator for circuits of the form $C(F_1, \ldots, F_m)$ where F_i's are polynomials of maximal sparsity s, and $\mathsf{trdeg}\{F_1, \ldots, F_m\} = k$.

Lemma 6.4. *Let $s, r, N \in \mathbb{N}$ and $m = r \log s + r \log r$. Consider the polynomial map*

$$\mathcal{G}^{\mathrm{BMS}}_{r,s}(y_1, \ldots, y_r, t_0, t_1, \ldots, t_n, w_1, \ldots, w_m, z_1, \ldots, z_m) := \mathcal{G}^{\mathrm{RC}}_{n,r}(\mathbf{y}, \mathbf{t}) + \mathcal{G}^{\mathrm{SSSV}}_{n,m}(\mathbf{w}, \mathbf{z}) .$$

Let $\mathbf{F} = \{F_1(\mathbf{X}), \ldots, F_m(\mathbf{X})\} \subseteq \mathbb{F}[\mathbf{X}]$ be a set of N-variate polynomials over $\mathbb{F}$ of sparsity at most s, such that $\mathsf{trdeg}\{F_1, \ldots, F_m\} \leq r$. Then for any polynomial of the form $G(\mathbf{x}) = C(F_1, \ldots, F_m)$, we have that $G \not\equiv 0$ if and only if

$$G \circ \mathcal{G}^{\mathrm{BMS}}_{r,s} \not\equiv 0 .$$

Furthermore, the generator $\mathcal{G}^{\mathrm{BMS}}_{r,s}$ is $\mathrm{poly}(r, \log s, n)$-$\Sigma\Pi\Sigma$ succinct.

Proof. By Lemma 6.3 and Proposition 4.2, it is enough to show that the map $\mathcal{G}^{\mathrm{SSSV}}_{n,m}(\mathbf{w}, \mathbf{z})$ preserves the rank of the Jacobian matrix. This follows from the fact that each $r \times r$ minor of this matrix is a polynomial of sparsity at most $r! \cdot s^r$ (since taking derivatives can only decrease the sparsity), and from Corollary 5.10.

The succinctness claim follows from Proposition 4.2 and Fact 5.6. ☐

Corollary 6.5. *There exists a $\mathrm{poly}(\log s, r, n)$-$\Sigma\Pi\Sigma$ succinct generator for the class of polynomials of the form $C(F_1, \ldots, F_m)$ such that each F_i has sparsity at most s and $\mathsf{trdeg}\{F_1, \ldots, F_m\} \leq r$.*

7 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 5.3 we have constructed a $\mathrm{poly}(\log w, \log n)$-$\Sigma\Pi\Sigma$ 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 $\mathrm{polylog}(w)$.

The definition of roABPs were given in Section 5.3. Throughout this section we assume that the ABP reads the variables in the order $X_1, X_2, \dots, X_N$. In Section 7.1 we give some short remarks regarding different variable orderings.

Our construction is based on the following generator by Forbes and Shpilka [37].

Lemma 7.1 (Forbes-Shpilka Generator for roABPs, Construction 3.13 in [37])**.** *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 $X_1, X_2, \dots, X_N$.*

Let $\omega \in \mathbb{F}$ be of multiplicative order at least $(Ndw^2)^2$, and $\beta_1, \dots, \beta_{w^2}$ be distinct elements of $\mathbb{F}$. Let $\{p_\ell \,:\, \ell \in [w^2]\}$ 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 $\mathbf{b} \in \{0,1\}^n$.

$$\mathcal{G}^{\mathrm{FS}}_{\mathbf{b}}(\mathbf{y}) = \sum_{\ell_1,\dots,\ell_n \in [w^2]} \prod_{i \in [n]} \Big((1-b_i) \cdot p_{\ell_{i-1}}(\omega^{\ell_i} y_i) + b_i \cdot p_{\ell_{i-1}}\big((\omega^{\ell_i} y_i)^{2^{i-1} d w^2}\big) \Big) \cdot p_{\ell_n}(y_{n+1}), \tag{7.1}$$

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

In [37] (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 $\mathbf{y} = \alpha$, there is a small roABP computing the polynomial whose coefficient vector is given by $(\mathcal{G}_{\mathbf{b}}(\alpha))_{\mathbf{b} \in \{0,1\}^n}$. In other words, for every choice of α, and associating $\mathbf{b}$ with a subset of $[n]$, we want a polynomial in $x_1, \dots, x_n$ such that the coefficient of $\mathbf{x}_b$ is $\mathcal{G}_{\mathbf{b}}(\alpha)$.

Definition 7.2 (Succinct Forbes-Shpilka Generator)**.** Let $n, w \in \mathbb{N}$, and ω, p_i's as in Lemma 7.1. Define

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

We first claim the the Forbes-Shpilka generator (7.1) is given by the coefficient vector of this polynomial.

Claim 7.3. *Assume the setup and notations of Definition 7.2. Then $\mathbf{coeff}_{\mathbf{x}}(P^{\mathrm{FS}}) = \mathcal{G}^{\mathrm{FS}}$.*

Proof. As explained earlier, we wish to show that the coefficient of $\mathbf{x}_b$ in the polynomial P^{FS} equals the $\mathbf{b}$-th coordinate of (7.1).

Fix a choice of $\ell_1,\ldots,\ell_n \in [w^2]$, and $\mathbf{b} \in \{0,1\}^n$. Consider the product

$$\prod_{i\in[n]} \left(p_{\ell_{i-1}}(\omega^{\ell_i} y_i) + x_i \cdot p_{\ell_{i-1}}((\omega^{\ell_i} y_i)^{2^{i-1}dw^2}) \right).$$

Since the product is over distinct variables, there is exactly one way to obtain the monomial $\mathbf{x}_b = \prod_{i:b_i=1} x_i$ in this product, and its coefficient will be

$$\prod_{i:b_i=1} p_{\ell_{i-1}}((\omega^{\ell_i} y_i)^{2^{i-1}dw^2}) \cdot \prod_{i:b_i=0} p_{\ell_{i-1}}(\omega^{\ell_i} y_i) \tag{7.2}$$

Finally, observe that (7.2) exactly equals

$$\prod_{i\in[n]} \left((1-b_i) \cdot p_{\ell_{i-1}}(\omega^{\ell_i} y_i) + b_i \cdot p_{\ell_{i-1}}((\omega^{\ell_i} y_i)^{2^{i-1}dw^2}) \right).$$

This is true for every fixed choice of $\ell_1,\ldots,\ell_n$, and the claim now follows from the linearity of the coefficients map. □

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

Claim 7.4. *For every setting $\mathbf{y} = \alpha$, the polynomial $P^{\mathrm{FS}}(\mathbf{x},\alpha)$ in Definition 7.2 can be computed by a width w^2 roABP in variable order $x_1, x_2 \ldots, x_n$.*

Proof. The construction is straightforward from Definition 7.2. Layer V_0 contains the source vertex s and layer V_{n+1} the sink vertex t. Layers $V_1,\ldots,V_n$ each contain w^2 vertices labeled by the set $[w^2]$. For every $i \in [n]$ and every $\ell \in V_i$, there is an edge from each vertex in the previous layer, labeled by the linear function (in x_i)

$$p_{\ell_{i-1}}(\omega^{\ell_i}\alpha_i) + x_i \cdot p_{\ell_{i-1}}((\omega^{\ell_i}\alpha_i)^{2^{i-1}dw^2}).$$

Finally, all vertices in V_n are connected to t with an edge labeled $p_{\ell_n}(\alpha_{n+1})$. □

Corollary 7.5. *The Forbes-Shpilka generator given in Lemma 7.1 is a width w^2-roABP succinct generator for degree d roABPs that read the variables in order $X_1, X_2, \ldots, X_N$.*

Proof. Immediate from Lemma 7.1, Claim 7.3 and Claim 7.4. □

7.1 Different variable orderings

The generator given by Forbes and Shpilka in Lemma 7.1 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 7.2 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 7.6. *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.*

8 Discussion and open problems

In this paper, 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 Meta-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 ([35, 5]), none of them seems to fit easily into the succinct setting, each for its own reasons.

For bounded-depth multilinear formulas, subexponential size hitting sets were obtained by Oliveira, Shpilka and Volk [78]. The construction there can be roughly described as hashing the N variables into $N^{1-\varepsilon}$ buckets, and then hitting each bucket independently using a generator for roABPs (in fact, commutative roABPs will suffice). The main challenge here seems to be the hashing part, which (in the succinct setting) would involve hashing monomials, and ensuring that the coefficient vector that is obtained through this process has a small circuit for any possible hash function.

The main technical tool which we do not know how to emulate in the succinct setting is the Klivans-Spielman [63] 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. Indeed, the noticeable difference from the constructions we have given in this paper is the *exponential* dependence on i in the exponent of t, a feature which is not clear how to emulate in the succinct setting.

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 succeeded 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 by 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 we are therefore currently unable to make these hitting sets succinct.

A particular interesting application of the Klivans and Spielman construction is in the recent work of Fenner, Gurjar and Thierauf [32] and its generalization by Gurjar and Thierauf [51]. These papers 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 [74], Raz [80], and Raz-Yehudayoff [83], is a read-once matrix, so 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 [83] 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 $\mathcal{C}$ where there are $\mathcal{C}$-succinct hitting sets for read-once determinants then this would show that no proof technique following the ideas of the above works can prove lower bounds for the class $\mathcal{C}$. 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 pseudorandom 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].

Acknowledgements

We thank Scott Aaronson, Andy Drucker, Josh Grochow, Mrinal Kumar, Shubhangi Saraf and Dor Minzer for useful conversations regarding this work. We also thank the anonymous reviewers for their careful reading of this paper and for many useful comments.

References

[1] SCOTT AARONSON: $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$. In *Open Problems in Mathematics*, pp. 1–122. Springer, 2016. [doi:10.1007/978-3-319-32162-2_1] 6

[2] SCOTT AARONSON AND ANDREW DRUCKER: Arithmetic natural proofs theory is sought, 2008. Shtetl-Optimized: The Blog of Scott Aaronson. 2, 4, 6, 12, 13, 18, 35

[3] SCOTT AARONSON AND AVI WIGDERSON: Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, 2009. Conference version in STOC'08. [doi:10.1145/1490270.1490272] 2, 13

[4] MANINDRA AGRAWAL: Proving lower bounds via pseudo-random generators. In *Proc. 25th Internat. Conf. on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05)*, pp. 92–105. Springer, 2005. [doi:10.1007/11590156_6] 9

[5] MANINDRA AGRAWAL, ROHIT GURJAR, ARPITA KORWAR, AND NITIN SAXENA: Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. [doi:10.1137/140975103, arXiv:1406.7535] 34

[6] MANINDRA AGRAWAL, CHANDAN SAHA, RAMPRASAD SAPTHARISHI, AND NITIN SAXENA: Jacobian hits circuits: Hitting sets, lower bounds for depth-D occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM J. Comput.*, 45(4):1533–1562, 2016. Conference version in STOC'12. [doi:10.1137/130910725, arXiv:1111.0582] 22, 29, 30, 31

[7] MANINDRA AGRAWAL, CHANDAN SAHA, AND NITIN SAXENA: Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proc. 45th STOC*, pp. 321–330. ACM Press, 2013. [doi:10.1145/2488608.2488649, arXiv:1209.2333] 24, 28

[8] MANINDRA AGRAWAL AND V. VINAY: Arithmetic circuits: A chasm at depth four. In *Proc. 49th FOCS*, pp. 67–75. IEEE Comp. Soc. Press, 2008. [doi:10.1109/FOCS.2008.32] 5

[9] MIKLÓS AJTAI: Σ_1^1-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983. [doi:10.1016/0168-0072(83)90038-6] 2, 3

[10] ERIC ALLENDER, JIA JIAO, MEENA MAHAJAN, AND V. VINAY: Non-commutative arithmetic circuits: depth reduction and size lower bounds. *Theoret. Comput. Sci.*, 209(1–2):47–86, 1998. [doi:10.1016/S0304-3975(97)00227-2] 5

[11] NOGA ALON AND RAVI B. BOPPANA: The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987. [doi:10.1007/BF02579196] 2

[12] NOGA ALON, ODED GOLDREICH, JOHAN HÅSTAD, AND RENÉ PERALTA: Simple construction of almost k-wise independent random variables. *Random Structures Algorithms*, 3(3):289–304, 1992. Conference version in FOCS'90. [doi:10.1002/rsa.3240030308] 13

[13] ALEKSANDR EGOROVICH ANDREEV: On a method for obtaining lower bounds for the complexity of individual monotone functions. *Doklady Akad. Nauk SSSR*, 282(5):1033–1037, 1985. Russian. English translation in Soviet Math. Dokl. 31 (1985) 530–534. 2

[14] THEODORE P. BAKER, JOHN GILL, AND ROBERT SOLOVAY: Relativizations of the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ question. *SIAM J. Comput.*, 4(4):431–442, 1975. [doi:10.1137/0204037] 2, 13

[15] WALTER BAUR AND VOLKER STRASSEN: The complexity of partial derivatives. *Theoret. Comput. Sci.*, 22(3):317–330, 1983. [doi:10.1016/0304-3975(83)90110-X] 5

[16] MALTE BEECKEN, JOHANNES MITTMANN, AND NITIN SAXENA: Algebraic independence and blackbox identity testing. *Inform. and Comput.*, 222:2–19, 2013. Conference version in ICALP'11. [doi:10.1016/j.ic.2012.10.004, arXiv:1102.2789] 30, 31

[17] STUART J. BERKOWITZ: On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18(3):147–150, 1984. [doi:10.1016/0020-0190(84)90018-8] 6

[18] MARKUS BLÄSER: Explicit tensors. In *Perspectives in Computational Complexity*, pp. 117–130. Springer, 2014. [doi:10.1007/978-3-319-05446-9_6] 9

[19] MARKUS BLÄSER, CHRISTIAN IKENMEYER, GORAV JINDAL, AND VLADIMIR LYSIKOV: Generalized matrix completion and algebraic natural proofs. In *Proc. 45th STOC*, pp. 1193–1206. ACM Press, 2018. [doi:10.1145/3188745.3188832] 17

[20] MARK BRAVERMAN: Polylogarithmic independence fools AC^0 circuits. *J. ACM*, 57(5):28:1–28:10, 2010. Conference version in CCC'09. [doi:10.1145/1754399.1754401] 13

[21] KARL BRINGMANN, CHRISTIAN IKENMEYER, AND JEROEN ZUIDDAM: On algebraic branching programs of small width. *J. ACM*, 65(5), 2018. Conference version in CCC'17. [doi:10.1145/3209663, arXiv:1702.05328] 16, 17

[22] PETER BÜRGISSER: *Completeness and Reduction in Algebraic Complexity Theory*. Springer, 2000. [doi:10.1007/978-3-662-04179-6] 19

[23] PETER BÜRGISSER, MICHAEL CLAUSEN, AND MOHAMMAD A. SHOKROLLAHI: *Algebraic Complexity Theory*. Springer, 1997. [doi:10.1007/978-3-662-03338-8] 9

[24] MARCO L. CARMOSINO, RUSSELL IMPAGLIAZZO, VALENTINE KABANETS, AND ANTONINA KOLOKOLOVA: Learning algorithms from natural proofs. In *Proc. 31st Conf. on Computational Complexity (CCC'16)*, pp. 10:1–10:24. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.10] 13

[25] BRYNMOR CHAPMAN AND R. RYAN WILLIAMS: The circuit-input game, natural proofs, and testing circuits with data. In *Proc. 6th Conf. on Innovations in Theoret. Computer Science (ITCS'15)*, pp. 263–270. ACM Press, 2015. [doi:10.1145/2688073.2688115] 4

[26] SURYAJITH CHILLARA, MRINAL KUMAR, RAMPRASAD SAPTHARISHI, AND V. VINAY: The chasm at depth four, and tensor rank : Old results, new insights. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2016. [ECCC:TR16-096, arXiv:1606.04200] 5

[27] TIMOTHY Y. CHOW: Almost-natural proofs. *J. Comput. System Sci.*, 77(4):728–737, 2011. Conference version in FOCS'08. [doi:10.1016/j.jcss.2010.06.017, arXiv:0805.1385] 4

[28] RICHARD A. DEMILLO AND RICHARD J. LIPTON: A probabilistic remark on algebraic program testing. *Inform. Process. Lett.*, 7(4):193–195, 1978. [doi:10.1016/0020-0190(78)90067-4] 7, 8, 9

[29] ZEEV DVIR AND AMIR SHPILKA: Locally Decodable Codes with two queries and Polynomial Identity Testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007. Conference version in STOC'05. [doi:10.1137/05063605X] 21

[30] ZEEV DVIR, AMIR SHPILKA, AND AMIR YEHUDAYOFF: Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. Conference version in STOC'08. [doi:10.1137/080735850] 9

[31] KLIM EFREMENKO, ANKIT GARG, RAFAEL OLIVEIRA, AND AVI WIGDERSON: Barriers for rank methods in arithmetic complexity. In *Proc. 9th Conf. on Innovations in Theoret. Computer Science (ITCS'18)*, pp. 1:1–1:19. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. [doi:10.4230/LIPIcs.ITCS.2018.1, arXiv:1710.09502] 17

[32] STEPHEN A. FENNER, ROHIT GURJAR, AND THOMAS THIERAUF: Bipartite perfect matching is in quasi-NC. In *Proc. 48th STOC*, pp. 754–763. ACM Press, 2016. [doi:10.1145/2897518.2897564, arXiv:1601.06319] 35

[33] MICHAEL A. FORBES: Deterministic divisibility testing via shifted partial derivatives. In *Proc. 56th FOCS*, pp. 451–465. IEEE Comp. Soc. Press, 2015. [doi:10.1109/FOCS.2015.35] 25, 26

[34] MICHAEL A. FORBES AND VENKATESAN GURUSWAMI: Dimension expanders via rank condensers. In *Proc. 19th Internat. Workshop on Randomization and Computation (RANDOM'15)*, pp. 800–814. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2015. [doi:10.4230/LIPIcs.APPROX-RANDOM.2015.800, arXiv:1411.7455] 20

[35] MICHAEL A. FORBES, RAMPRASAD SAPTHARISHI, AND AMIR SHPILKA: Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proc. 46th STOC*, pp. 867–875. ACM Press, 2014. [doi:10.1145/2591796.2591816] 20, 27, 28, 34

[36] MICHAEL A. FORBES AND AMIR SHPILKA: On identity testing of tensors, low-rank recovery and compressed sensing. In *Proc. 44th STOC*, pp. 163–172. ACM Press, 2012. [doi:10.1145/2213977.2213995, arXiv:1111.0663] 20

[37] MICHAEL A. FORBES AND AMIR SHPILKA: Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proc. 54th FOCS*, pp. 243–252. IEEE Comp. Soc. Press, 2013. [doi:10.1109/FOCS.2013.34, arXiv:1209.2408] 6, 32

[38] MICHAEL A. FORBES, AMIR SHPILKA, IDDO TZAMERET, AND AVI WIGDERSON: Proof complexity lower bounds from algebraic circuit complexity. In *Proc. 31st Conf. on Computational Complexity (CCC'16)*, pp. 32:1–32:17. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.CCC.2016.32, arXiv:1606.05050] 26

[39] MICHAEL A. FORBES, AMIR SHPILKA, AND BEN LEE VOLK: Succinct hitting sets and barriers to proving algebraic circuits lower bounds. In *Proc. 49th STOC*, pp. 653–664. ACM Press, 2017. [doi:10.1145/3055399.3055496, arXiv:1701.05328] 1

[40] HERVÉ FOURNIER, NUTAN LIMAYE, GUILLAUME MALOD, AND SRIKANTH SRINIVASAN: Lower bounds for depth-4 formulas computing Iterated Matrix Multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015. Conference version in STOC'14. [doi:10.1137/140990280] 5

[41] MERRICK L. FURST, JAMES B. SAXE, AND MICHAEL SIPSER: Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. Conference version in FOCS'81. [doi:10.1007/BF01744431] 2, 3

[42] ARIEL GABIZON AND RAN RAZ: Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008. Conference version in FOCS'05. [doi:10.1007/s00493-008-2259-3] 15, 20

[43] ODED GOLDREICH, SHAFI GOLDWASSER, AND SILVIO MICALI: How to construct random functions. *J. ACM*, 33(4):792–807, 1986. Conference version in FOCS'84. [doi:10.1145/6490.6503] 3, 13

[44] JOSHUA A. GROCHOW: Unifying known lower bounds via Geometric Complexity Theory. *Comput. Complexity*, 24(2):393–475, 2015. Conference version in CCC'14. [doi:10.1007/s00037-015-0103-x] 2, 4, 6, 17

[45] JOSHUA A. GROCHOW, MRINAL KUMAR, MICHAEL SAKS, AND SHUBHANGI SARAF: Towards an algebraic natural proofs barrier via polynomial identity testing. *ECCC*, 2017. [ECCC:TR17-009, arXiv:1701.01717] 5, 11

[46] JOSHUA A. GROCHOW, KETAN D. MULMULEY, AND YOUMING QIAO: Boundaries of VP and VNP. In *Proc. 43rd Internat. Colloq. on Automata, Languages and Programming (ICALP'16)*, pp. 34:1–34:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016. [doi:10.4230/LIPIcs.ICALP.2016.34, arXiv:1605.02815] 17

[47] ANKIT GUPTA, PRITISH KAMATH, NEERAJ KAYAL, AND RAMPRASAD SAPTHARISHI: Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. Conference version in CCC'13. [doi:10.1145/2629541] 5, 6, 9, 17

[48] ANKIT GUPTA, PRITISH KAMATH, NEERAJ KAYAL, AND RAMPRASAD SAPTHARISHI: Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Conference version in FOCS'13. [doi:10.1137/140957123] 5

[49] ROHIT GURJAR, ARPITA KORWAR, AND NITIN SAXENA: Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *Theory of Computing*, 13(2):1–21, 2017. Conference version in CCC'16. [doi:10.4086/toc.2017.v013a002, arXiv:1601.08031] 27

[50] ROHIT GURJAR, ARPITA KORWAR, NITIN SAXENA, AND THOMAS THIERAUF: Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complexity*, 26(4):835–880, 2017. Conference version in CCC'15. [doi:10.1007/s00037-016-0141-z, arXiv:1411.7341] 25, 26

[51] ROHIT GURJAR AND THOMAS THIERAUF: Linear matroid intersection is in quasi-NC. In *Proc. 49th STOC*, pp. 821–830. ACM Press, 2017. [doi:10.1145/3055399.3055440] 35

[52] JURIS HARTMANIS AND RICHARD E. STEARNS: On the computational complexity of algorithms. *Trans. Amer. Math. Soc.*, 117:285–306, 1965. [doi:10.2307/1994208] 2

[53] JOHAN HÅSTAD: Almost optimal lower bounds for small depth circuits. In *Proc. 18th STOC*, pp. 6–20. ACM Press, 1986. [doi:10.1145/12130.12132] 2, 3

[54] JOHAN HÅSTAD, RUSSELL IMPAGLIAZZO, LEONID A. LEVIN, AND MICHAEL LUBY: A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999. [doi:10.1137/S0097539793244708] 3, 13

[55] JOOS HEINTZ AND CLAUS-PETER SCHNORR: Testing polynomials which are easy to compute (Extended Abstract). In *Proc. 12th STOC*, pp. 262–272. ACM Press, 1980. [doi:10.1145/800141.804674] 9

[56] VALENTINE KABANETS AND RUSSELL IMPAGLIAZZO: Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complexity*, 13(1–2):1–46, 2004. Conference version in STOC'03. [doi:10.1007/s00037-004-0182-6] 9, 14

[57] ZOHAR SHAY KARNIN AND AMIR SHPILKA: Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011. Conference version in CCC'08. [doi:10.1007/s00493-011-2537-3] 21

[58] RICHARD M. KARP AND RICHARD J. LIPTON: Turing machines that take advice. *L'Enseignement Mathématique*, 28(2):191–209, 1982. [doi:10.5169/seals-52237] 2

[59] NEERAJ KAYAL: An exponential lower bound for the sum of powers of bounded degree polynomials. *Electron. Colloq. on Comput. Complexity (ECCC)*, 2012. [ECCC:TR12-081] 5, 6, 9

[60] NEERAJ KAYAL, NUTAN LIMAYE, CHANDAN SAHA, AND SRIKANTH SRINIVASAN: An exponential lower bound for homogeneous depth four arithmetic circuits. *SIAM J. Comput.*, 46(1):307–335, 2017. Conference version in FOCS'14. [doi:10.1137/151002423] 5

[61] NEERAJ KAYAL, CHANDAN SAHA, AND RAMPRASAD SAPTHARISHI: A super-polynomial lower bound for regular arithmetic formulas. In *Proc. 46th STOC*, pp. 146–153. ACM Press, 2014. [doi:10.1145/2591796.2591847] 5

[62] NEERAJ KAYAL AND SHUBHANGI SARAF: Blackbox polynomial identity testing for depth 3 circuits. In *Proc. 50th FOCS*, pp. 198–207. IEEE Comp. Soc. Press, 2009. [doi:10.1109/FOCS.2009.67] 21

[63] ADAM R. KLIVANS AND DANIEL A. SPIELMAN: Randomness efficient identity testing of multivariate polynomials. In *Proc. 33rd STOC*, pp. 216–223. ACM Press, 2001. [doi:10.1145/380752.380801] 15, 29, 34

[64] PASCAL KOIRAN: Arithmetic circuits: The chasm at depth four gets wider. *Theoret. Comput. Sci.*, 448:56–65, 2012. [doi:10.1016/j.tcs.2012.03.041] 5

[65] MATTHIAS KRAUSE AND STEFAN LUCKS: Pseudorandom functions in TC^0 and cryptographic limitations to proving lower bounds. *Comput. Complexity*, 10(4):297–313, 2001. [doi:10.1007/s000370100002] 3

[66] MRINAL KUMAR AND SHUBHANGI SARAF: On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.*, 46(1):336–387, 2017. Conference version in FOCS'14. [doi:10.1137/140999335, arXiv:1404.1950] 5

[67] CARSTEN LUND, LANCE FORTNOW, HOWARD J. KARLOFF, AND NOAM NISAN: Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. Conference version in FOCS'90. [doi:10.1145/146585.146605] 2

[68] MEENA MAHAJAN AND V. VINAY: A combinatorial algorithm for the determinant. In *Proc. 8th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'97)*, pp. 730–738. ACM Press, 1997. ACM DL. 6

[69] RAGHU MEKA AND DAVID ZUCKERMAN: Pseudorandom generators for polynomial threshold functions. *SIAM J. Comput.*, 42(3):1275–1301, 2013. Conference version in STOC'10. [doi:10.1137/100811623, arXiv:0910.4122] 13

[70] KETAN D. MULMULEY: Geometric Complexity Theory V: Equivalence between blackbox derandomization of Polynomial Identity Testing and derandomization of Noether's normalization lemma. *J. Amer. Math. Soc.*, 30:225–309, 2017. Conference version in FOCS'12. [doi:10.1090/jams/864, arXiv:1209.5993] 12

[71] KETAN D. MULMULEY AND MILIND SOHONI: Geometric Complexity Theory I: An approach to the $\mathsf{P} \stackrel{?}{=} \mathsf{NP}$ and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001. [doi:10.1137/S009753970038715X] 16

[72] JOSEPH NAOR AND MONI NAOR: Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. Conference version in STOC'90. [doi:10.1137/0222053] 13

[73] MONI NAOR AND OMER REINGOLD: Number-theoretic constructions of efficient pseudorandom functions. *J. ACM*, 51(2):231–262, 2004. Conference version in FOCS'97. [doi:10.1145/972639.972643] 3

[74] NOAM NISAN: Lower bounds for non-commutative computation. In *Proc. 23rd STOC*, pp. 410–418. ACM Press, 1991. [doi:10.1145/103418.103462] 4, 5, 6, 18, 35

[75] NOAM NISAN: Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991. [doi:10.1007/BF01375474] 13

[76] NOAM NISAN AND AVI WIGDERSON: Hardness vs randomness. *J. Comput. System Sci.*, 49(2):149–167, 1994. Conference version in FOCS'88. [doi:10.1016/S0022-0000(05)80043-1] 13

[77] NOAM NISAN AND AVI WIGDERSON: Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity*, 6(3):217–234, 1996. Conference version in FOCS'95. [doi:10.1007/BF01294256] 4, 5, 6

[78] RAFAEL OLIVEIRA, AMIR SHPILKA, AND BEN LEE VOLK: Subexponential size hitting sets for bounded depth multilinear formulas. *Comput. Complexity*, 25(2):455–505, 2016. Conference version in CCC'15. [doi:10.1007/s00037-016-0131-1, arXiv:1411.7492] 34

[79] RAN RAZ: Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. Conference version in FOCS'04. [doi:10.4086/toc.2006.v002a006] 4, 6

[80] RAN RAZ: Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009. Conference version in STOC'04. [doi:10.1145/1502793.1502797] 5, 35

[81] RAN RAZ: Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(7):135–177, 2010. Conference version in STOC'08. [doi:10.4086/toc.2010.v006a007] 19

[82] RAN RAZ AND AMIR YEHUDAYOFF: Balancing syntactically multilinear arithmetic circuits. *Comput. Complexity*, 17(4):515–535, 2008. [doi:10.1007/s00037-008-0254-0] 5

[83] RAN RAZ AND AMIR YEHUDAYOFF: Lower bounds and separations for constant depth multilinear circuits. *Comput. Complexity*, 18(2):171–207, 2009. Conference version in CCC'08. [doi:10.1007/s00037-009-0270-8] 4, 6, 35

[84] ALEXANDER ALEXANDROVICH RAZBOROV: Lower bounds on the monotone complexity of some Boolean functions. *Dokl. Akad. Nauk SSSR*, 281(4):798–801, 1985. Available at author's webpage. 2, 4

[85] ALEXANDER ALEXANDROVICH RAZBOROV: 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):333–338, 1987. [doi:10.1007/BF01137685] 2, 3

[86] ALEXANDER ALEXANDROVICH RAZBOROV AND STEVEN RUDICH: Natural proofs. *J. Comput. System Sci.*, 55(1):24–35, 1997. Conference version in STOC'94. [doi:10.1006/jcss.1997.1494] 2, 3, 4, 6, 7, 8, 10, 13

[87] RAMPRASAD SAPTHARISHI: *Unified Approaches to Polynomial Identity Testing and Lower Bounds*. Ph. D. thesis, Chennai Mathematical Institute, 2012. Available at author's webpage. 22, 30

[88] RAMPRASAD SAPTHARISHI: A survey of lower bounds in arithmetic circuit complexity, 2016. Available at Github. 5

[89] NITIN SAXENA: Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009. [ECCC:TR09-101] 10

[90] NITIN SAXENA: Progress on polynomial identity testing - II. In *Perspectives in Computational Complexity*, pp. 131–146. Springer, 2014. [doi:10.1007/978-3-319-05446-9_7, arXiv:1401.0976] 10

[91] NITIN SAXENA AND C. SESHADHRI: An almost optimal rank bound for depth-3 identities. *SIAM J. Comput.*, 40(1):200–224, 2011. Conference version in CCC'09. [doi:10.1137/090770679] 21

[92] NITIN SAXENA AND C. SESHADHRI: Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012. Conference version in STOC'11. [doi:10.1137/10848232, arXiv:1011.3234] 21

[93] NITIN SAXENA AND C. SESHADHRI: From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33:1–33:33, 2013. Conference version in FOCS'10. [doi:10.1145/2528403] 21

[94] JACOB T. SCHWARTZ: Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. [doi:10.1145/322217.322225] 7, 8, 9

[95] ADI SHAMIR: $\mathsf{IP} = \mathsf{PSPACE}$. *J. ACM*, 39(4):869–877, 1992. Conference version in FOCS'90. [doi:10.1145/146585.146609] 2

[96] AMIR SHPILKA AND ILYA VOLKOVICH: Read-once polynomial identity testing. *Comput. Complexity*, 24(3):477–532, 2015. Conference version in STOC'08. [doi:10.1007/s00037-015-0105-8] 15, 23

[97] AMIR SHPILKA AND AMIR YEHUDAYOFF: Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. [doi:10.1561/0400000039] 2, 6, 10, 11, 21

[98] 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, 3

[99] VOLKER STRASSEN: Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. *Numerische Mathematik*, 20(3):238–251, 1973. [doi:10.1007/BF01436566] 5

[100] ÉVA TARDOS: The gap between monotone and non-monotone circuit complexity is exponential. *Combinatorica*, 8(1):141–142, 1988. [doi:10.1007/BF02122563] 2

[101] SÉBASTIEN TAVENAS: Improved bounds for reduction to depth 4 and depth 3. *Inform. and Comput.*, 240:2–11, 2015. Conference version in MFCS'13. [doi:10.1016/j.ic.2014.09.004, arXiv:1304.5777] 5

[102] LESLIE G. VALIANT: Completeness classes in algebra. In *Proc. 11th STOC*, pp. 249–261. ACM Press, 1979. [doi:10.1145/800135.804419] 18

[103] LESLIE G. VALIANT, SVEN SKYUM, S. BERKOWITZ, AND CHARLES RACKOFF: Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in MFCS'81. [doi:10.1137/0212043] 5, 18

[104] R. RYAN WILLIAMS: Nonuniform ACC circuit lower bounds. *J. ACM*, 61(1):2:1–2:32, 2014. Conference version in CCC'11. [doi:10.1145/2559903] 4

[105] R. RYAN WILLIAMS: Natural proofs versus derandomization. *SIAM J. Comput.*, 45(2):497–529, 2016. Conference version in STOC'13. [doi:10.1137/130938219, arXiv:1212.1891] 4, 10, 13

[106] ANDREW CHI-CHIH YAO: Separating the polynomial-time hierarchy by oracles (Preliminary version). In *Proc. 26th FOCS*, pp. 1–10. IEEE Comp. Soc. Press, 1985. [doi:10.1109/SFCS.1985.49] 2, 3

[107] RICHARD ZIPPEL: Probabilistic algorithms for sparse polynomials. In *Proc. Internat. Symp. on Symbolic and Algebraic Computation (EUROSAM'79)*, pp. 216–226. Springer, 1979. [doi:10.1007/3-540-09519-5_73] 7, 8, 9

AUTHORS

Michael A. Forbes
Assistant professor
University of Illinois at Urbana-Champaign, IL, USA
miforbes@illinois.edu
http://miforbes.cs.illinois.edu/

Amir Shpilka
Professor
Tel Aviv University, Tel Aviv, Israel
shpilka@post.tau.ac.il
http://www.cs.tau.ac.il/~shpilka

Ben Lee Volk
Postdoctoral scholar
California Institute of Technology
Pasadena, CA
benleevolk@gmail.com
http://www.its.caltech.edu/~benleevo/

ABOUT THE AUTHORS

MICHAEL A. FORBES obtained his Ph. D. in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 2014, where he was co-advised by Scott Aaronson and Amir Shpilka. His dissertation developed new deterministic algorithms to solve cases of the polynomial identity testing problem. In 2017, he joined the faculty of University of Illinois at Urbana-Champaign. His research focuses on interaction of randomness, algebra, and computation.

AMIR SHPILKA obtained his Ph. D. in Computer Science and Mathematics from the Hebrew University in Jerusalem in 2001 under the supervision of Avi Wigderson. From 2005 to 2014 he was a faculty member at the CS department at the Technion. In October 2014 he joined the CS department at Tel-Aviv University. His research interests lie in complexity theory, especially in algebraic complexity.

BEN LEE VOLK is a postdoctoral scholar at Caltech. He graduated from Tel-Aviv University, advised by Amir Shpilka. Previously, he obtained a B. Sc. and an M. Sc. at the Technion. His research interests involve algebraic complexity theory, and other noun phrases which contain the words "algebra" and "complexity."