Towards an algebraic natural proofs barrier via polynomial identity testing

Joshua A. Grochow[*], Mrinal Kumar[†],
Michael Saks[‡], and Shubhangi Saraf[§]

January 9, 2017

Abstract

We observe that a certain kind of algebraic proof—which covers essentially all known algebraic circuit lower bounds to date—cannot be used to prove lower bounds against VP if and only if what we call succinct hitting sets exist for VP. This is analogous to the Razborov–Rudich natural proofs barrier in Boolean circuit complexity, in that we rule out a large class of lower bound techniques under a derandomization assumption. We also discuss connections between this algebraic natural proofs barrier, geometric complexity theory, and (algebraic) proof complexity.

1 Introduction

The natural proofs barrier [51] showed that a large class of circuit-based proof techniques could not separate P from NP, assuming the existence of pseudo-random generators of a certain strength. In light of the recent advances in techniques for lower bounds on algebraic circuits [5–7, 11, 15, 20, 21, 23–42, 47, 52, 55], it is natural to wonder whether our current algebraic techniques could plausibly separate VP from VNP, or whether there is some barrier in this setting as well. People often hand-wave about an "algebraic natural proofs barrier," by analogy to Razborov–Rudich, but it has not been clear what this means precisely, and to date no such barrier is known in a purely algebraic setting (see below for a discussion of the related work by Aaronson and Drucker [2, 3] in a partially algebraic, partially Boolean setting).

There are several difficulties in coming up with such a barrier in the algebraic context. Razborov and Rudich's notion of natural proof has two key ingredients: (1) largeness—the technique works to show that random functions are hard—and (2) constructivity—deciding whether a function satisfies the hypotheses of the technique can be done efficiently given its truth table. These two ingredients in combination allow them to make the connection to pseudo-random generators. However, in the algebraic world, all *three* of these notions are unclear: What should largeness mean in an algebraic context? What should constructivity

[*] Department of Computer Science, University of Colorado, Boulder, CO, USA and the Santa Fe Institute, Santa Fe, NM, USA, joshua.grochow@cs.colorado.edu

[†] Department of Computer Science, Rutgers University, Piscataway, NJ, USA, mrinal.kumar@rutgers.edu

[‡] Department of Mathematics, Rutgers University, Piscataway, NJ, USA, saks@math.rutgers.edu

[§] Departments of Computer Science & Mathematics, Rutgers University, Piscataway, NJ, USA, shubhangi.saraf@rutgers.edu

mean? Is there a good algebraic notion of pseudo-random generator?[1] In a mixed Boolean-algebraic setting, Aaronson and Drucker [2,3] provide satisfactory answers, but in a purely algebraic setting finding a constellation of three answers to these questions that align to give a satisfying algebraic natural proofs barrier has been an open question for more than twenty years.

In the purely algebraic setting—algebraic circuits over an arbitrary field—we take largeness to mean Zariski-openness (the complement of the zero set of a set of polynomial equations), and constructivity to mean that the property is computable by an algebraic circuit whose size is polynomial in the number of coefficients of the function being tested. These two properties cover essentially all known algebraic lower bounds to date [2,18] (see also [53, Section 3.9]). Rather than connecting these notions directly to PRGs, we connect them to a slightly different derandomization problem, but one that is natural from the algebraic viewpoint: polynomial identity testing (PIT). Here, we suggest that the coefficient vectors of random linear projections of the determinant (respectively, a generic algebraic circuit) should produce good hitting sets for restricted versions of PIT. (This is closely related to Aaronson and Drucker's suggestion that they form a pseudo-random family of algebraic functions [2,3]; see Section 1.2 for details.) We observe that if this is true, then many proof techniques—including those of the recent advances—cannot be used to separate VP_{ws} from VNP. As in the original natural proofs barrier, we thus show that a strong enough derandomization assumption implies that certain techniques cannot prove strong lower bounds. We recently learned that Forbes, Shpilka and Volk came to the same connection with PIT independently, and were able to show some of the derandomization assumptions unconditionally [14].

In the final two sections, we comment on how this algebraic natural proofs barrier bears on geometric complexity theory, and how it might be used to prove lower bounds in (algebraic) proof complexity.

1.1 The idea

Almost all algebraic circuit lower bounds to date proceed either by the substitution method, or by a "rank-type" method, namely: associate to each polynomial f some matrix $M(f)$—e. g., a matrix of partial derivatives or shifted partial derivatives, perhaps exponentially large—show an upper bound on the rank of $M(f)$ for any $f \in \mathcal{C}_{easy}$, and show a lower bound on the rank of $M(f_{hard})$ for some polynomial f_{hard}. In all examples to date, the entries of the matrix $M(f)$ are linear functions of the coefficients of f; as the rank of $M(f)$ is determined by the vanishing of its minors, we can view this method as an instance of the following more general "polynomial method." For a polynomial f, let $\mathrm{coeff}(f)$ denote its coefficient vector. The polynomial method is to find a "meta-polynomial" T (called "test polynomials" in [18])—whose variables are the coefficients of polynomials f—such that $T(\mathrm{coeff}(f)) = 0$ for all $f \in \mathcal{C}_{easy}$, but $T(\mathrm{coeff}(f_{hard})) \neq 0$.

The first step here—as in the original natural proofs barrier—is to consider the circuit complexity of the meta-polynomials T themselves, relative to their number of inputs. The new idea here is to consider which classes of meta-polynomials have $\{\mathrm{coeff}(f) : f \in \mathcal{C}_{easy}\}$ as a hitting set.

[1] Although Agrawal's notion of algebraic PRG [4] is useful in its own setting, it is not clear whether it could be used for an algebraic natural proofs barrier, and in fact connecting our formulation to Agrawal's PRGs remains an interesting open problem.

Let us consider what this looks like for the rank-type methods mentioned above. Suppose that we are considering polynomial families $f = (f_n)_{n=1,2,3\dots}$ in n^c variables of degree n. The space of such polynomials has dimension $\binom{n+n^c-1}{n} = 2^{\Theta(n \log n)}$. Since this will be the number of variables of our meta-polynomials (we might call them "meta-variables"), let us denote it by N. The matrices $M(f)$ typically have dimension poly(N), which is still $2^{\Theta(n \log n)}$. If we are considering whether or not $M(f)$ has rank $\leq r$ or $> r$, then we are considering the (non)vanishing of the $(r+1) \times (r+1)$ minors, which are themselves determinants of size at most poly$(N) \times$ poly(N). Therefore, these meta-polynomials lie in the circuit class we denote $\mathsf{VP}_{ws}(N)$, which is defined just like VP_{ws}, but where everything—degree, circuit size, etc.—is measured as a function of the number of variables N. This circuit complexity upper bound on the meta-polynomials is the algebraic analogue of Razborov and Rudich's constructivity criterion.

Now, suppose we want to prove a lower bound against some class $\mathcal{C}_{easy}$ using such a rank-type argument. If the coefficient vectors of polynomials in $\mathcal{C}_{easy}$ form a hitting set (perhaps infinite) for $\mathsf{VP}_{ws}(N)$, then no meta-polynomial as in the preceding paragraph can vanish on $\mathcal{C}_{easy}$, precluding such arguments. This is the fundamental connection we advance between algebraic circuit lower bounds (by the polynomial method) and polynomial identity testing.

1.2 Relationship with previous work

Efremenko, Landsberg, Schenck, and Weyman [12, 13] proved unconditionally that the method of shifted partial derivatives cannot prove a lower bound stronger than $\Omega(n^2)$ on the permanent versus determinant problem. While parts of their methods are not specific to these polynomials, their results *are* specific to the method of shifted partial derivatives. In contrast, our general framework has the potential to rule out proving lower bounds by *any* method where the meta-polynomials are easily computable. While in this paper all our results are conditional, some of them are made unconditional in Forbes–Shpilka–Volk [14].

Aaronson and Drucker [2, 3] (see Aaronson's survey [1, Section 6.5.3] for an overview) had similar ideas, but ours differ in several respects. One strength of their work compared to ours is that they considered not just algebraic, but mixed Boolean-algebraic settings—that is, considering polynomials over finite fields as Boolean functions of the bitwise description of the field elements—and this allowed them to draw equivalences between the existence of Boolean and (suitably formulated) algebraic pseudo-random functions. In contrast, our work is purely algebraic, and rather than using pseudo-random generators, we use hitting sets for polynomial identity testing.

The difference between their work and ours which allows us to make the connection with PIT is as follows. They considered a polynomial family f_n to be pseudo-random if it could not be distinguished from a random polynomial family of similar degree by any meta-polynomial computed by small circuits C_n, in the sense that $\Pr[f_n(C_n(f_n(x))) = f_n(x)]$ was negligible as a function of n (smaller than $1/n^c$ for any c) [3, Slide 8]. In order for this to make sense, they considered polynomial families (f_n) over fields of growing size $\mathbb{F}_{p(n)}$ (and the probability is taken uniformly over $x \in \mathbb{F}_{p(n)}^n$). This is quite close to the usual Boolean definition of pseudo-randomness, which is what allowed them to make that connection. In contrast, we say that a meta-polynomial computed by some circuit C_n distinguishes one polynomial f_n from another polynomial g_n if C_n outputs 0 when given the coefficient vector of f_n as input, and outputs a nonzero value when given the coefficient vector of g_n. (By using interpolation, we can replace "coefficient vector" with "vector of evaluations at sufficiently many points" for any class which supports interpolation, that is, which is closed under affine linear transformations of the variables.) By only considering a meta-polynomial to distinguish one polynomial from another by its vanishing/non-vanishing, rather than in the probabilistic sense of pseudo-randomness, we are able to work over arbitrary fields, and make the connection with PIT instead of pseudo-random functions.

We note, however, that if in their work one instead considers algebraic Turing machines (a la Blum–Shub–Smale) to distinguish functions—as they suggest at one point—then probability goes away. By considering the possible paths through such a machine, one gets a condition which is a logical combination of conditions on the vanishing/non-vanishing of certain polynomials, rather than the vanishing/non-vanishing of a single polynomial. See Remark 1 for more details.

2 Preliminaries

A *family* of polynomials $f = (f_n)$ consists of one polynomial f_n for each n, usually on a number of variables that depends on n. A sequence of integers $a_1, a_2, \dots$ is *p-bounded* if there is a polynomial $n^c + c$ such that $a_n \leq n^c + c$ for all sufficiently large n. A *p-family* is a family of polynomials (f_n) such that the number of variables of f_n and the degree of f_n are both p-bounded. We will primarily be interested in p-families throughout.

A non-uniform algebraic complexity class is a collection of families of polynomials. VP is the collection of p-families (f_n) such that f_n computable by an algebraic circuit of poly(n) size. VNP is the collection of p-families (g_n) such that there is a family $(f_n(x,e)) \in \mathsf{VP}$ such that $g_n(x) = \sum_{e \in \{0,1\}^{\text{poly}(n)}} f_n(x,e)$. VP_{ws} is the collection of p-families $f = (f_n)$ such that $f_n(x) = \det_{\text{poly}(n)}(L_n(x))$ where $L_n(x)$ is matrix whose entires are affine linear functions of the x_i. $\Sigma\Pi\Sigma$ is the collection of p-families computable by polynomial-size, depth-three, layered circuits, with a linear combination gate at the output, preceded by a layer of multiplication gates, preceded by a layer of linear combinations of the input; that is, the polynomial is a sum of polynomially many products of linear functions of the inputs.

A polynomial $f(x_1, \dots, x_n)$ is a *projection* of a polynomial $g(y_1, \dots, y_m)$ if there are affine linear functions $\ell_1(\vec{x}), \dots, \ell_m(\vec{x})$ such that $f(\vec{x}) = g(\ell_1(\vec{x}), \dots, \ell_m(\vec{x}))$, identically as polynomials. A polynomial family $f = (f_n)$ is a *p-projection* of a polynomial family $g = (g_n)$ if there is a polynomial $t(n)$ such that for all n, f_n is a projection of $g_{t(n)}$.

Let $\text{Poly}^d(v)$ denote the space of *homogeneous* polynomials of degree d in v variables, and $\text{Poly}^{\leq d}(v)$ denote the space of (not necessarily homogeneous) polynomials of degree at most d in v variables. Homogeneity is used for technical simplicity; essentially everything we say can be modified to several other natural settings, such as non-homogeneous polynomials or multilinear polynomials.

Rather than the definitional viewpoint of a complexity class as a collection of families of polynomials, it will be useful to "reverse the order of quantifiers", and to consider, for each n, a subset of $\text{Poly}^{d(n)}(v(n))$, and to consider a complexity class as a family of such subsets, one for each n. This viewpoint is implicit in much work on lower bounds in algebraic complexity theory, going back to work of Strassen (e. g., [54]), and is explicit in geometric complexity theory (e. g., [10, 43–46]). An example will help make this clear: In terms of lower bounds showing that some polynomial is not in VP_{ws}, it is useful to think of VP_{ws} as being "captured" by the following family of sets:

$$\mathcal{D}_n \overset{def}{=} \{f(x) \in \text{Poly}^{\leq n}(n^2) : (\exists L)[f(x) = \det_n(L(x))]\}$$

where the L we consider here are those such that $L(X)$ is an $n \times n$ matrix whose entries are affine linear combinations of the variables x_i. The family $\mathcal{D} = (\mathcal{D}_n)_{n=1,2,3,\ldots}$ captures VP_{ws} in the sense that, given a family of polynomials $g = (g_n)$, showing that $g_n \notin \mathcal{D}_{m_n}$ for all polynomially bounded sequences (m_n) and for infinitely many n proves that $g \notin \mathsf{VP}_{ws}$. We crystalize this into the following definition:

Definition 1. A family of subsets $(\mathcal{F}_n)$ with $\mathcal{F}_n \subseteq \mathrm{Poly}^{d(n)}(v(n))$ *captures* a non-uniform algebraic complexity class $\mathcal{C}$ if:

1. For every family of polynomials (f_n) with $f_n \in \mathcal{F}_n$ for all n, it follows that $(f_n) \in \mathcal{C}$, and

2. For every family of polynomials $f = (f_n) \in \mathcal{C}$, there is a polynomially bounded sequence of integers m_n such that $f_n \in \mathcal{F}_{m_n}$ for every n.

We say that $(\mathcal{F}_n)$ *captures* $\mathcal{C}$ *with padding* if we replace the last item by

2′. For every family of polynomials $f = (f_n) \in \mathcal{C}$, there are polynomially bounded sequences of integers e_n, m_n and a family of linear forms $\ell = (\ell_n)_{n=1}^{\infty}$ such that $\ell_n(x)^{e_n} f_n(x) \in \mathcal{F}_{m_n}$ for every n.

It is readily seen that the family $\mathcal{D}_n$ above captures VP_{ws}. Its homogeneous version,

$$\mathcal{D}_n^h \overset{def}{=} \{f(x) \in \mathrm{Poly}^n(n^2) : (\exists L)[f(x) = \det_n(L(x))]\}$$

where we only conisder *linear* L (zero constant term), captures homogeneous polynomials in VP_{ws} with padding.

Similarly, the family

$$\mathcal{SPS}_n \overset{def}{=} \left\{ f \in \mathrm{Poly}^{\leq n}(n) : [\exists a_{ijk} \in \mathbb{F}] \left(f = \sum_{i=1}^{n} \prod_{j=1}^{d(i)} \sum_{k=1}^{n} a_{ijk} x_k \right) \right\}$$

captures $\Sigma\Pi\Sigma$. While essentially all non-uniform algebraic complexity classes that are ever considered have a natural family of sets that captures them, note that such families are not unique. For example, VP_{ws} is also captured by the family of sets

$$\mathcal{W}_n \overset{def}{=} \{f \in \mathrm{Poly}^{\leq n}(n) : f \text{ can be computed by a weakly-skew circuit of size } \leq n\}.$$

While each $\mathcal{W}_n$ is quite different from each $\mathcal{D}_n$, this merely reflects the fact that weakly-skew circuit size and determinantal complexity are not equal, despite the fact that they are polynomially related.

Throughout, whenever we refer to a complexity class such as $\mathsf{VP}_{ws}(n)$, we really mean "$\mathcal{F}_n$, for any fixed family $\mathcal{F}_n$ that captures VP_{ws} (possibly with padding)."

2.1 Meta-polynomials and meta-complexity classes

Given a space of polynomials $\mathrm{Poly}^{d_n}(v_n)$, we may consider *meta-polynomials* on this space, which are polynomials T whose variables correspond to the *coefficients* of polynomials in $\mathrm{Poly}^{d_n}(v_n)$. That is, T is a polynomial in $N = \binom{d_n+v_n-1}{d_n}$ variables. We denote the space of homogeneous meta-polynomials of degree D by $\mathrm{Poly}^D(\mathrm{Poly}^{d_n}(v_n)) \cong \mathrm{Poly}^D(N)$. Given a polynomial $f \in \mathrm{Poly}^{d_n}(v_n)$ and a meta-polynomial $T \in \mathrm{Poly}(\mathrm{Poly}^{d_n}(v_n))$, we denote by $T(\mathrm{coeff}(f))$ the evaluation of T at the coefficient vector of f. We will generally use capital letters to denote meta-polynomials, their degrees, and their number of variables, and lower-case letters for (non-meta) polynomials.

Example. The familiar polynomial $b^2 - 4ac$ may be considered as a meta-polynomial on the space $\mathrm{Poly}^2(2)$ of degree 2 homogeneous polynomials in 2 variables, namely, $\mathrm{Poly}^2(2) = \{ax^2 + bxy + cy^2 : a, b, c \in \mathbb{F}\}$. Then $T = b^2 - 4ac \in \mathrm{Poly}^2(\mathrm{Poly}^2(2))$, and evaluating T at a polynomial $f = ax^2 + bxy + cy^2$ has the usual and natural meaning.

We will want to consider families of meta-polynomials $T = (T_n)$ with $T_n \in \mathrm{Poly}(\mathrm{Poly}^{d_n}(v_n))$. If v_n, d_n are themselves at least linear in n, then the number of variables of T_n is exponential in n, so this family does not technically fit into the usual algebraic complexity classes as defined above. We would nonetheless like an analogue of the above classes where T_n may depend on more than $\mathrm{poly}(n)$ variables, but its other relevant quantities are polynomial in its (usually much larger than $\mathrm{poly}(n)$) number of variables. We annotate such classes with a capital N, where N_n is the number of variables of T_n (in this case, $\dim \mathrm{Poly}^{d_n}(v_n)$).

Definition 2 (Stretched complexity classes)**.** Given an algebraic complexity class $\mathcal{C}$, and a function $N(n)$, we define *$\mathcal{C}$ with stretch N*, denoted $\mathcal{C}(N)$, as the class of families of polynomials $T = (T_n)$ such that there is a family $\overline{T} \in \mathcal{C}$ with $T_n = \overline{T}_{N(n)}$.

For most standard algebraic complexity classes, such as VP, VP_{ws}, VNP, or $\Sigma\Pi\Sigma$, this is equivalent to:

Definition 2′ (Alternative definition of stretched, for standard classes)**.** Given an algebraic complexity class $\mathcal{C}$, and a function $N(n)$, we define *$\mathcal{C}$ with stretch N*, denoted $\mathcal{C}(N)$, as the class of families of polynomials $T = (T_n)$ such that T satisfies the hypotheses of $\mathcal{C}$ with "polynomial in n" everywhere replaced by "polynomial in $N(n)$."

To see that the two are equivalent: Given $T \in \mathcal{C}(N)$ according to Definition 2′, if we let $\overline{n}(N)$ be the inverse of $N(n)$ rounded to the nearest integer, then defining a family $\overline{T}_n = T_{\overline{n}(N(n))}$ satisfies Definition 2. The opposite direction is clear.

For example, $\mathsf{VP}(N)$ denotes the class of families of polynomials $T = (T_n)$ where T_n has $\mathrm{poly}(N)$ many variables, is of $\mathrm{poly}(N)$ degree, and can be computed by circuits of $\mathrm{poly}(N)$ size. We define $\mathsf{VNP}(N)$, $\mathsf{VP}_{ws}(N)$ and $\Sigma\Pi\Sigma(N)$ analogously.

Since we will typically be considering polynomials in $\mathrm{poly}(n)$ many variables with $\mathrm{poly}(n)$ degree, the space $\mathrm{Poly}^{d_n}(v_n)$ will have dimension $N_n = 2^{n^{O(1)}}$, so we have that $n = \mathrm{poly}(\log N)$.

3 An algebraic natural proofs barrier via polynomial identity testing

We start by giving our definition of "algebraic natural property;" an algebraic natural proof in our sense will essentially be one that uses such a property. As is the case with Razborov–Rudich natural proofs, the latter is not a precise, formal definition, but in practice this will cause us no difficulties, and in particular does not affect our results (which are precise and formal). In the algebraic setting, a property of polynomials is a collection of subsets $C_n \subseteq \mathrm{Poly}^{d_n}(v_n)$ for some (usually p-bounded) sequences d_n, v_n. (Recall that everything we say is easily adapted to other kinds of polynomials such as non-homogeneous or multilinear.)

Definition 3 (Natural property)**.** A property of polynomials $C = (C_n)$ with $C_n \subseteq \mathrm{Poly}^{d_n}(v_n)$ is *natural* if it contains a set $C_n^* \subseteq C_n$ for each n satisfying the following two conditions:

1. *Largeness:* C_n^* is the complement of the zero-set of a meta-polynomial T_n.

2. *Constructivity:* The meta-polynomial family $T = (T_n)$ has degree and circuit size bounded by a polynomial in the number of its variables ($= \mathrm{poly}(\dim \mathrm{Poly}^{d_n}(v_n)) = \mathrm{poly}(\binom{d_n+v_n-1}{d_n})$). That is, $T \in \mathsf{VP}(N)$ for $N_n = \dim \mathrm{Poly}^{d_n}(v_n)$.

3. *Usefulness:* The algebraic circuit size of any family of functions (f_n) with $f_n \in C_n$ for all n is super-polynomial, that is, for any constant d, for sufficiently large n the circuit size of f_n is greater than n^d.

Remark 1 (Deciding by (non)vanishing)**.** It is important for the connection with PIT that constructivity here be in terms of computing T symbolically as a polynomial (or at least, some T' such that $\{f : T'(f) \neq 0\} \subseteq C_n$), and not merely in terms of *deciding* whether a given function f is contained in C_n^* (as is the case with Razborov–Rudich natural proofs).

However, we note that even if we had allowed instead, say, Blum–Shub–Smale-style algebraic Turing machines to decide, given coeff(f), whether or not $f \in C_n^*$, then much of the machinery still survives. In particular, the generic path through a BSS machine is still Zariski-open, being the intersection of finitely many Zariski-open subsets, and the "yes/no" output of the machine on generic inputs depends only on the vanishing/non-vanishing of a given polynomial. However, we would then need our hitting set to hit not only this final "decider" polynomial, but also all of the "branching" polynomials encountered along the generic computation path. If we wanted to consider all paths through the BSS machine, and not just the generic one, the situation becomes significantly more complicated, and as far as we are aware hitting sets for such computations have not been considered in the literature.

Remark 2 (Choice of field)**.** In terms of which fields to work over, in order to make the connection with derandomization, we want to work over fields that are large enough that derandomizing PIT over those fields is at least plausible. For simplicity, it may be easier to think of $\mathbb{F}$ as any infinite field. In principle, one could also work over a family of fields $\mathbb{F}_{s(n)}$ of size $s(n)$ greater than twice the degree of the polynomials under consideration (so the Schwarz–Zippell-DeMillo–Lipton Lemma holds). Note that for $s(n) < 2^{\mathrm{poly}(n)}$, the algebraic natural proofs barrier of Aaronson and Drucker also applies [2, 3].

We generalize this to:

Definition 4 (Γ-natural against Λ)**.** For two complexity classes Γ, Λ, a property $C = (C_n)$ is Γ-*natural against* Λ if it contains a subset $C_n^* \subseteq C_n$ satisfying:

1. *Largeness:* C_n^* is the complement of the zero-set of a meta-polynomial T_n.

2. Γ-*Constructivity:* The meta-polynomial family $T = (T_n)$ is in the meta-complexity class $\Gamma(N)$, where $N_n = \dim \mathrm{Poly}^{d_n}(v_n)$.

3. *Usefulness against* Λ*:* Any family of functions $f = (f_n)$ with $f_n \in C_n$ for all n is not contained in Λ.

As observed in [18], essentially all known algebraic circuit lower bounds to date are natural in this sense; in fact, most of them are VP_{ws}-natural against the relevant complexity class, as they are defined by the rank of a matrix of size $\mathrm{poly}(N) \times \mathrm{poly}(N)$ (see Section 1.1).

The key observation is the following. If there is a hitting set against VP which consists of the coefficient vectors of polynomials of number of variables, degree, and size $\mathrm{poly}(\log n)$, then there is no property that is VP-natural against VP. In other words, if for every meta-polynomial $T \in \mathsf{VP}(N)$, there is some polynomial $f \in \mathsf{VP}$ such that $T(\mathrm{coeff}(f)) \neq 0$, then one cannot prove a lower bound against VP by exhibiting a meta-polynomial that vanishes on a family of sets capturing VP. As all such lower bounds to date are of this form [18], and it is reasonable to expect future such lower bounds to be as well (see, e. g., [54] or [18, Appendix B] for a more extended discussion of this expectation), this rules out quite a large class of lower bounds methods.

We formalize this observation with a definition and a theorem:

Definition 5 (Succinct hitting set)**.** An algebraic complexity class Λ is a *succinct hitting set* against another class Γ if there is a family of sets $\Lambda(n)$ which captures Λ, such that $\{\mathrm{coeff}(f) : f \in \Lambda(n)\}$ is a hitting set against $\Gamma(N)$, where N is the dimension of the ambient space of $\Lambda(n)$. Namely, for all nonzero $T \in \Gamma(N)$, there is some $f \in \Lambda(n)$ such that $T(\mathrm{coeff}(f)) \neq 0$.

Theorem 1. *For any two algebraic complexity classes Γ, Λ, there is a Λ-succinct hitting set against Γ if and only if there is no property which is Γ-natural against Λ.*

We have essentially already given the proof in the paragraph above. □

The main open question is thus:

Open Question 1. *Is VP a succinct hitting set against VP? Is VP_{ws} a succinct hitting set against VP_{ws}?*

We note that it is not even obvious whether or not VNP is a succinct hitting set against VP_{ws}. An important first step would be to show that known hitting sets against subclasses $\Gamma \subseteq \mathsf{VP}$ can be made Λ-succinct for as small a class Λ as possible. For several pairs (Λ, Γ) this is achieved in [14].

Remark 3 (Generators)**.** A *generator* for a class Γ is a vector-valued function $\vec{G}(x_1, \ldots, x_s)$ such that for any nonzero $f \in \Gamma$, $f(\vec{G}(\vec{x}))$ is not identically zero as a polynomial in $\vec{x}$. In other words, the image of $\vec{G}$—essentially an s-dimensional variety—is a hitting set (perhaps infinite) against Γ. The number of variables, s, is called the seed length of the generator; generators of small seed length are useful because they reduce PIT for Γ from a many-variable problem to s-variable PIT, which is easily solved for small s. For most standard classes Λ, we note that if Λ is a succinct hitting set against Γ, then this set is a generator against Γ of small seed length. For most classes—such as $\mathsf{VP}, \mathsf{VNP}, \mathsf{VP}_{ws}$, $\mathsf{\Sigma\Pi\Sigma}$—are the image of a simply specified polynomial map $\vec{G}$ on few parameters. For example, the set of linear projections of the $n \times n$ determinant captures $\mathsf{VP}_{ws}(n)$ (with padding). This means that we may consider $\mathsf{VP}_{ws}(n)$ as the image of the map $M_{n^2 \times n^2} \to \mathrm{Poly}^n(n^2)$ which sends an $n^2 \times n^2$ matrix L to the function $\det_n(L(\vec{x}))$, where we think of the $n \times n$ matrix x simply as a vector of length n^2. If VP_{ws} is a hitting set for some class $\Gamma(N)$, then we may view it as a generator for $\Gamma(N)$ using the preceding encoding. The seed length of this generator is $n^4 = \mathrm{poly}(n)$ variables, but it outputs vectors in $\mathrm{Poly}^n(n^2)$, which has dimension N that is exponential in n. So when VP_{ws} is a hitting set against some Γ, this generator still reduces from finding a hitting set in $N = 2^{\Theta(n \log n)}$ variables to finding a hitting set in $n^4 = \mathrm{poly}(\log N)$ variables. As in the preceding example of VP_{ws} and the determinant, generators of small seed length are obvious for many classes; for VP this is somewhat less obvious, but is still true [50].

4 Relationship with other topics in complexity

4.1 Geometric complexity theory

In geometric complexity theory (GCT), the suggestion is not merely to use the polynomial method to find a meta-polynomial T that vanishes on $\mathcal{C}_{easy}$ but not on some f_{hard}, but to additionally take advantage of the fact that most standard non-uniform classes $\mathcal{C}(n)$ are invariant under the action of some nontrivial group G, such as GL_n or S_n. This is because most measures of complexity do not depend on how we name the variables (leading to S_n symmetry), and in many cases only change polynomially given a linear change of variables (leading to GL_n symmetry). The suggestion, without loss of generality, is thus to use a property C_n to separate $\mathcal{C}(n)$ from f_{hard} such that C_n is also sent to itself by the same symmetry group. In this case, rather than considering a single meta-polynomial T, we may, again without loss of generality, consider the entire linear span V of all meta-polynomials T' that are in the G-orbit of T. (When G is S_n it is clear that V is finite-dimensional; even over infinite fields, however, this is also true of the GL_n-orbit of T.) V is then a representation of G or G-module; following [18] we refer to a G-module of meta-polynomials as a "test G-module," since its vanishing is a test for having a given G-invariant property.

For $G = \mathrm{GL}_n$ (a natural group of symmetries for many standard algebraic circuit classes such as VP, VP_{ws}, VNC, VNP, VQP, $\Sigma\Pi\Sigma$), every irreducible G-module contains an essentially unique highest weight vector (see, e. g., [16]) (=highest weight test polynomial), which is an "HWV obstruction" in the terminology of [8]. (Conversely, every HWV obstruction gives rise to a test module.) Considering these HWV obstructions directly, Bürgisser and Ikenmeyer were able to prove lower bounds on matrix multiplication using the technology of GCT [8]. This raises the natural question of: given the label λ of an irreducible GL_n-module (λ is a partition with at most n parts, see, e. g., [16]), how computationally hard is it to construct its (unique) highest weight test polynomial? However, from the viewpoint of algebraic natural proofs, we are led to a related but slightly different question.

The first natural question to think of is to determine the circuit complexity of the HWV obstructions. However, algebraic natural proofs suggests asking something still further.

Namely, suppose that Γ-natural proofs cannot prove lower bounds against Λ, and suppose that $\Lambda(n)$ is invariant under a group G_n (not necessarily GL_n—in particular, we do not need the theory of highest weights for what we are about to say). Then given a sequence of test G_n-modules V_n, potentially useful against $\Lambda(n)$, if there is a sequence of meta-polynomials $T_n \in V_n$ such that $(T_n)_{n=1,2,3,\ldots}$ is in Γ, then for infinitely many n, V_n is not useful against $\Lambda(n)$—that is, V_n does not vanish identically on $\Lambda(n)$. We are thus led to the question:

Open Question 2. *For any given sequence of test G-modules V_n, what is the complexity of the* easiest *family of meta-polynomials ($T_n \in V_n$)?*

In particular, while the complexity of any given $T_n \in V_n$ doesn't change within the orbit of T_n, V_n itself contains all *linear combinations* of points on this orbit, and some such linear combinations could have significantly lower complexity than, say, the HWVs in V_n (when V_n is a test GL_n-module).

Note that, whether or not there is a natural proofs barrier for VP, the above question is interesting. For if there is such a barrier, then any family of test G-modules with low-complexity polynomials cannot be used to prove lower bounds.[2] Conversely, if there is no

[2] It is interesting to note that if, for a given sequence of labels $\lambda(n)$, we could find an upper bound on the

such barrier, then any family of test G-modules with low-complexity polynomials might be a good place to look for test polynomials to prove lower bounds, since we might hope that low-complexity test polynomials would be easier to understand and therefore easier to use to try to prove lower bounds.

This question is perhaps more immediately interesting in the following specific cases: Given a class Γ for which it is shown in [14] that Γ-natural proofs cannot prove lower bounds against VP, which families $V_{\lambda(n)}$ of test GL_n-modules contain a family of test polynomials T_n such that $(T_n) \in \Gamma$? Note that, even for test GL_n-modules V, the highest weight meta-polynomials need not be the easiest polynomials in T. So although considering HWV obstructions may be useful for proving lower bounds, in order to prove that certain test GL_n-modules are *not* useful for lower bounds, one needs to consider the more general Open Question 2.

4.2 Algebraic proof complexity

Pitassi [48, 49] and Grochow & Pitassi [19] introduced the Ideal Proof System (IPS), for refuting unsatisfiable CNFs using algebraic reasoning. While IPS is a very strong proof system—at least as strong as Extended Frege—they also introduced a variant of this system called the *Geometric* IPS (it is an open question whether Geometric IPS can p-simulate general IPS). Using the connection in this paper it may be plausible to prove unconditional lower bounds against Geometric IPS. We now discuss this in a bit more detail.

Definition (Geometric Ideal Proof System, "Geometric IPS," [19, Appendix B])**.** Given an unsatisfiable system of polynomial equations $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$, a *geometric IPS certificate* of unsatisfiability consists of an algebraic circuit $C(y_1, \ldots, y_m)$ such that

1. $C(\vec{0}) = 1$, and
2. $C(f_1(\vec{x}), \ldots, f_m(\vec{x})) = 0$, in other words, C is a polynomial relation amongst the f_i.

For any algebraic circuit class $\mathcal{C}$, a *geometric $\mathcal{C}$-IPS proof* is an algebraic circuit in $\mathcal{C}$ on inputs $y_1, \ldots, y_m$ computing some geometric IPS certificate.

This system may be used to prove that a 3CNF formula is unsatisfiable as follows. Given a 3CNF formula with m clauses, we translate it into a system of m polynomials of degree at most 3 in the natural way, so that any Boolean assignment to the variables satisfies a clause iff the corresponding polynomial evaluates to 0. Then the 3CNF is unsatisfiable iff the corresponding equations $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0$ are unsatisfiable over $\mathbb{F}$. In [19, Appendix B] it shown that geometric IPS, without any complexity bounds on the circuit computing a certificate, is a sound and complete proof system for such systems of equations. In fact, over any algebraically closed field or any dense subfield of $\mathbb{C}$, they showed that the same is true even if the equations $x_i^2 - x_i = 0$ are omitted.

The idea of the geometric IPS is to consider the equations $f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n)$ as a map $f \colon \mathbb{F}^n \to \mathbb{F}^m$, and to note that the system of equations $f_1 = \ldots = f_m = 0$ is satisfiable iff 0 is in the image of the map f. A geometric IPS certificate proves that, not only

easiest family of test polynomials in *any* family of test GL_n-modules isomorphic to $V_{\lambda(n)}$, then this could be used to rule out *multiplicity* obstructions. At the moment, there are essentially no techniques known for ruling out multiplicity obstructions, only for ruling out occurrence obstructions, e. g., [9, 17, 22].

is 0 not in the image, but 0 is not even in the *closure* of the image of the map f. The geometric object of interest here is thus the image of the map f.

Suppose we have a family $(\mathcal{F}_n)_{n=1,2,\dots}$ of systems of polynomial equations

$$\mathcal{F}_n = (f_{n,1}(x_1, \dots, x_{n^c}), \dots, f_{n,n^d}(\vec{x})),$$

such that the images of the maps $f_n \colon \mathbb{F}^{n^c} \to \mathbb{F}^{n^d}$ are a hitting set against some circuit class Λ. Then, by condition (2) of the above definition, no geometric Λ-IPS certificate can exist. Although here we are using the evaluations of polynomials rather than their coefficient vectors, note that for any class Λ capable of interpolation—that is, closed under affine linear transformations—a succinct hitting set can be defined either in terms of coefficient vectors or in terms of the vector of evaluations at sufficiently many points.

Open Question 3. *For various Λ for which hitting sets are known, prove lower bounds on the Geometric Λ-Ideal Proof System by finding a succinct hitting set of the following form: there is a family of unsatisfiable 3CNFs (φ_n) such that, if f_n is the above polynomial map associated to φ_n, then the image of f_n is a hitting set against Λ.*

Of course, it would also be interesting to show that for certain Λ no hitting sets of this form exist.

Unfortunately, we were unable to get the same connection to work for general IPS. The natural object to look at for general IPS is not the image of f, but rather its graph $\{(\vec{\alpha}, \vec{f}(\vec{\alpha})) : \vec{\alpha} \in \mathbb{F}^n\}$. The issue is that, when the f_i are themselves described by small circuits, as is essentially always the case in instances of complexity-theoretic interest, the function $y_i - f_i(\vec{x})$ is a very easily computable function which vanishes on the graph of f.

Open Question 4. *Find and exploit an analogous connection between algebraic natural proofs / hitting sets and (general) IPS.*

Acknowledgments

We thank Scott Aaronson and Andy Drucker for conversations about their work [2,3], the relationship between the two approaches and how they might be combined. We thank Amir Shpilka for conversations related to Section 4.1 and for his encouragement to publicize our thinking, even in light of the results of [14] which (independently) supercede ours. During the course of this work, J.A.G. was supported by A. Borodin's NSERC Grant # 482671, an Omidyar Fellowship from the Santa Fe Institute, and NSF grant DMS-1620484; M.S. was supported by NSF grant CCF-1218711 and by Simons Foundation Award 332622; and S.S. was supported in part by NSF grant CCF-1350572.

References

[1] Scott Aaronson. $\mathsf{P} \overset{?}{=} \mathsf{NP}$. In John Forbes Nash, Jr. and Michael Th. Rassias, editors, *Open Problems in Mathematics*, pages 1–122. Springer, 2016. Updated version available at `http://www.scottaaronson.com/papers/pnp.pdf`.

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

[3] Scott Aaronson and Andrew Drucker. Impagliazzo's worlds in arithmetic complexity. Talk presented at the Workshop on Complexity and Cryptography: Status of Impagliazzo's Worlds, Center for Computational Intractability, Princeton, NJ, June 5, 2009, 2009. Slides available at `http://www.scottaaronson.com/talks/arith.ppt`.

[4] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of software technology and theoretical computer science*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105. Springer, Berlin, 2005. `doi:10.1007/11590156_6`.

[5] 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. Originally appeared in STOC '12; preprint available as arXiv:1111.0582 [cs.CC] and ECCC Tech. Report TR11-143. `doi:10.1137/130910725`.

[6] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *FOCS '08: 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75. IEEE Computer Society, 2008. `doi:10.1109/FOCS.2008.32`.

[7] Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. In *CCC '16: 31st IEEE Conference on Computational Complexity*, pages 30:1–30:25, 2016. Preprint available as arXiv:1511.07136 [cs.CC] and ECCC Tech. Report TR15-184. `doi:10.4230/LIPIcs.CCC.2016.30`.

[8] Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. In *STOC '13: 45th Annual ACM Symposium on Theory of Computing*, pages 141–150. ACM, New York, 2013. Preprint available as arXiv:1210.8368 [cs.CC]. `doi:10.1145/2488608.2488627`.

[9] Peter Bürgisser, Christian Ikenmeyer, and Greta Panova. No occurrence obstructions in geometric complexity theory. In *FOCS '16: 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 386–395, 2016. Preprint of full version available as arXiv:1604.06431. `doi:10.1109/FOCS.2016.49`.

[10] Peter Bürgisser, J. M. Landsberg, Laurent Manivel, and Jerzy Weyman. An overview of mathematical issues arising in the Geometric Complexity Theory approach to $\mathsf{VP} \neq \mathsf{VNP}$. *SIAM J. Comput*, 40(4):1179–1209, 2011. `doi:10.1137/090765328`.

[11] Suryajith Chillara, Mrinal Kumar, Ramprasad Saptharishi, and V. Vinay. The chasm at depth four, and tensor rank : Old results, new insights. arXiv:1606.04200 [cs.CC] and ECCC Tech. Report TR16-096, 2016.

[12] Klim Efremenko, J.M. Landsberg, Hal Schenck, and Jerzy Weyman. On minimal free resolutions and the method of shifted partial derivatives in complexity theory. arXiv:1504.05171 [cs.CC], 2015.

[13] Klim Efremenko, J.M. Landsberg, Hal Schenck, and Jerzy Weyman. The method of shifted partial derivatives cannot separate the permanent from the determinant. arXiv:1609.02103 [math.AG], 2016.

[14] Michael Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving algebraic circuits lower bounds. ECCC Tech. Report TR17-007, 2017.

[15] Michael A. Forbes, Mrinal Kumar, and Ramprasad Saptharishi. Functional lower bounds for arithmetic circuits and connections to boolean circuit complexity. In *CCC '16: 31st IEEE Conference on Computational Complexity*, pages 33:1–33:19, 2016. Preprint available as arXiv:1605.04207 [cs.CC] and ECCC Tech. Report TR16-045. doi:10.4230/LIPIcs.CCC.2016.33.

[16] William Fulton and Joe Harris. *Representation theory*, volume 129 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1991.

[17] Fulvio Gesmundo, Christian Ikenmeyer, and Greta Panova. Geometric complexity theory and matrix powering. arXiv:1611.00827, 2016.

[18] Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *computational complexity*, 24:393–475, 2015. Special issue from IEEE CCC 2014. Open access. doi:10.1007/s00037-015-0103-x.

[19] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing. In *FOCS '14: 55th Annual IEEE Symposium on Foundations of Computer Science*, 2014. Preprint of full version available as arXiv:1404.3820 [cs.CC] and ECCC Tech. Report TR14-052. Submitted for journal publication. doi: 10.1109/FOCS.2014.20.

[20] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. Assoc. Comput. Mach.*, 61(6):33:1–33:16, 2014. Originally appeared in CCC '13; preprint available as ECCC Tech. Report TR12-098. doi: 10.1145/2629541.

[21] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput*, 45(3):1064–1079, 2016. Originally appeared in FOCS '13; preprint available as ECCC Tech. Report TR13-026. doi: 10.1137/140957123.

[22] Christian Ikenmeyer and Greta Panova. Rectangular Kronecker coefficients and plethysms in geometric complexity theory. In *FOCS '16: 57th Annual IEEE Symposium on Foundations of Computer Science*, pages 396–405, 2016. Preprint of full version available as arXiv:1512.03798. doi:10.1109/FOCS.2016.50.

[23] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. ECCC Tech. Report TR12-081, 2012.

[24] Neeraj Kayal. Arithmetic circuit complexity (tutorial). In *STACS '14: 31st Annual Symposium on Theoretical Aspects of Computer Science*, pages 28–28, 2014. doi: 10.4230/LIPIcs.STACS.2014.28.

[25] Neeraj Kayal, Pascal Koiran, Timothée Pecatte, and Chandan Saha. Lower bounds for sums of powers of low degree univariates. In *ICALP '15: 42nd International Colloquium on Automata, Languages and Programming*, pages 810–821, 2015. doi: 10.1007/978-3-662-47672-7_66.

[26] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *FOCS '14: 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 61–70, 2014. Preprint available as ECCC Tech. Report TR14-005. doi:10.1109/FOCS.2014.15.

[27] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. Superpolynomial lower bounds for depth-4 homogeneous arithmetic formulas. In *STOC '14: 46th Annual ACM Symposium on Theory of Computing*, pages 119–127, 2014. doi:10.1145/2591796.2591823.

[28] Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth three circuits. In *STACS '16: 33rd Annual Symposium on Theoretical Aspects of Computer Science*, pages 46:1–46:15, 2016. Preprint available as ECCC Tech. Report TR15-154. doi:10.4230/LIPIcs.STACS.2016.46.

[29] Neeraj Kayal and Chandan Saha. Lower bounds for sums of products of low arity polynomials. ECCC Tech. Report TR15-073, 2015.

[30] Neeraj Kayal and Chandan Saha. Multi-k-ic depth three circuit lower bound. In *STACS '15: 32nd Annual Symposium on Theoretical Aspects of Computer Science*, pages 527–539, 2015. Preprint available as ECCC Tech. Report TR15-015. doi:10.4230/LIPIcs.STACS.2015.527.

[31] Neeraj Kayal and Chandan Saha. Lower bounds for depth-three arithmetic circuits with small bottom fanin. *Computational Complexity*, 25(2):419–454, 2016. Originally appeared in CCC '15; preprint available as ECCC Tech. Report TR14-089. doi:10.1007/s00037-016-0132-0.

[32] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *STOC '14: 46th Annual ACM Symposium on Theory of Computing*, pages 146–153, 2014. Preprint available as ECCC Tech. Report TR13-091. doi:10.1145/2591796.2591847.

[33] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In *ICALP '16: 43rd International Colloquium on Automata, Languages and Programming*, pages 33:1–33:15, 2016. Preprint available as ECCC Tech. Report TR16-006. doi:10.4230/LIPIcs.ICALP.2016.33.

[34] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In *STOC '16: 48th Annual ACM Symposium on Theory of Computing*, pages 626–632, 2016. Preprint available as ECCC Tech. Report TR15-181. doi:10.1145/2897518.2897550.

[35] Pascal Koiran. Arithmetic circuits: the chasm at depth four gets wider. *Theoret. Comput. Sci.*, 448:56–65, 2012. Preprint available as arXiv:1006.4700 [cs.CC]. doi:10.1016/j.tcs.2012.03.041.

[36] Mrinal Kumar, Gaurav Maheshwari, and Jayalal Sarma. Arithmetic circuit lower bounds via maximum-rank of partial derivative matrices. *TOCT*, 8(3):8, 2016. Originally appeared in ICALP '13; preprint available as arXiv:1302.3308 [cs.CC] and ECCC Tech. Report TR13-023. doi:10.1145/2898437.

[37] Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. arXiv:1507.00177 [cs.CC] and ECCC Tech. Report TR15-109, 2015.

[38] Mrinal Kumar and Ramprasad Saptharishi. Finer separations between shallow arithmetic circuits. ECCC Tech. Report TR16-137, 2016.

[39] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *FOCS '14: 55th Annual IEEE Symposium on Foundations of Computer Science*, pages 364–373, 2014. Preprint available as arXiv:1404.1950 [cs.CC] and ECCC Tech. Report TR14-045. `doi:10.1109/FOCS.2014.46`.

[40] Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: It's all about the top fan-in. *SIAM J. Comput.*, 44(6):1601–1625, 2015. Originally appeared in STOC '14; preprint available as arXiv:1302.3308 [cs.CC] and ECCC Tech. Report TR13-028. `doi:10.1137/140999220`.

[41] Mrinal Kumar and Shubhangi Saraf. Arithmetic circuits with locally low algebraic rank. In *CCC '16: 31st IEEE Conference on Computational Complexity*, pages 34:1–34:27, 2016. Preprint available as ECCC Tech. Report TR15-194. `doi:10.4230/LIPIcs.CCC.2016.34`.

[42] Mrinal Kumar and Shubhangi Saraf. Sums of products of polynomials in few variables: Lower bounds and polynomial identity testing. In *CCC '16: 31st IEEE Conference on Computational Complexity*, pages 35:1–35:29, 2016. Preprint available as arXiv:1504.06213 [cs.CC] and ECCC Tech. Report TR15-071. `doi:10.4230/LIPIcs.CCC.2016.35`.

[43] Ketan D. Mulmuley. On P vs. NP and Geometric Complexity Theory. *J. Assoc. Comput. Mach.*, 58(2):5:1–5:26, 2011. `doi:10.1145/1944345.1944346`.

[44] Ketan D. Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *J. Amer. Math. Soc.*, 30(1):225–309, 2017. Extended abstract appeared in FOCS '12. `doi:10.1090/jams/864`.

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

[46] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput*, 38(3):1175–1206, 2008. `doi:10.1137/080718115`.

[47] Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic Independence over Positive Characteristic: New Criterion and Applications to Locally Low Algebraic Rank Circuits. In *MFCS '16: 41st Symposium on Mathematical Foundations of Computer Science*, volume 58 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 74:1–74:15, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.MFCS.2016.74`.

[48] Toniann Pitassi. Algebraic propositional proof systems. In *Descriptive Complexity and Finite Models, Proceedings of the DIMACS Workshop held at Princeton University,*

Princeton, NJ, January 14‘17, 1996. Edited by Neil Immerman and Phokion G. Kolaitis, volume 31 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 215–244. American Mathematical Society, 1996.

[49] Toniann Pitassi. Propositional proof complexity and unsolvability of polynomial equations. In *Proceedings of the International Congress of Mathematicians. Vol. III. Sections 10–19. Held in Berlin, August 18-27, 1998*, pages 215–244, 1998.

[50] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory Comput.*, 6:135–177, 2010. Extended abstract appeared in STOC ’08. `doi:10.4086/toc.2010.v006a007`.

[51] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. System Sci.*, 55(1, part 1):24–35, 1997. `doi:10.1006/jcss.1997.1494`.

[52] Shubhangi Saraf. Recent progress on lower bounds for arithmetic circuits. In *CCC ’14: 29th IEEE Conference on Computational Complexity*, pages 155–160, 2014. `doi:10.1109/CCC.2014.23`.

[53] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: a survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388 (2010), 2009. `doi:10.1561/0400000039`.

[54] Volker Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM J. Comput*, 3:128–149, 1974. `doi:10.1137/0203010`.

[55] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. In *MFCS ’13: 38th Symposium on Mathematical Foundations of Computer Science*, pages 813–824, 2013. Preprint available as arXiv:1304.5777 [cs.CC]. `doi:10.1007/978-3-642-40313-2_71`.