Guest Column: Algebraic Natural Proofs[1]

Ben Lee Volk[2]

Abstract

Algebraic Natural Proofs is a recent framework which formalizes the type of reasoning used for proving most lower bounds on algebraic computational models. This concept is similar to and inspired by the famous natural proofs notion of Razborov and Rudich [RR97] for boolean circuit lower bounds, but, unlike in the boolean case, it is an open problem whether this constitutes a barrier for proving super-polynomial lower bounds for strong models of algebraic computation. From an algebraic-geometric viewpoint, it is also related to basic questions in Geometric Complexity Theory (GCT), and from a meta-complexity theoretic viewpoint, it can be seen as an algebraic version of the MCSP problem. We survey the recent work around this concept which provides some evidence both for and against the existence of an algebraic natural proofs barrier, with an emphasis on the different viewpoints and the connections to other areas.

1 Introduction

Computational complexity theory studies the limits of efficient computation. Computational complexity theorists, however, are as much as obsessed with studying their own limitations. A series of introspective results, often called "barriers", shows that many of the successful techniques of the field are likely to be unable to succeed in tackling its major open problems, such as P vs. NP. A few of the most famous examples are the relativization barrier of Baker, Gill, and Solovay [BGS75], which covers diagonalization techniques, such as those used in the proof of the time hierarchy theorem; the natural proofs barrier of Razborov and Rudich [RR97], which covers the circuit complexity methods which show that $\mathsf{AC}^0 \subsetneq \mathsf{NC}^1$; and the algebrization barrier of Aaronson and Wigderson [AW09], which covers arithmetization based methods such as those used in the proof of $\mathsf{IP} = \mathsf{PSPACE}$.

These results, and the respective barriers, all apply for *boolean* models of computation, such as Turing machines and boolean circuits. There is, however, a parallel universe of *algebraic* complexity, which studies the limitations of efficient algorithms for computing algebraic problems using algebraic computational models (we give a background on algebraic complexity in Section 2). Algebraic computation seems more structured, and hence potentially easier to analyze, than boolean computation. Nevertheless, many of the basic open problems for algebraic computation, such as the algebraic analog of the P vs. NP question, are as widely open as their boolean counterparts.

[2] School of Computer Science, Reichman University (IDC Herzliya), Israel. benleevolk@gmail.com. Part of this work was written while at UT Austin, supported by NSF Grant CCF-1705028.

Up until recently, however, we had no excuses for this slow progress, namely, no barrier results for algebraic complexity.

Since the main computational model in algebraic complexity is the non-uniform algebraic circuit model (see Section 2 for definitions), the most relevant barrier among the list mentioned above is the natural proofs barrier of Razborov and Rudich [RR97]. The main insight of Razborov and Rudich was that most of the existing lower bound proofs against restricted circuit classes do more than proving lower bounds: they provide *efficient algorithms* that distinguish random functions from functions with small circuits. They call proofs with these properties "natural proofs", and observe that under the cryptographic assumption that pseudorandom function families exist, such proofs can't work for more powerful classes of circuits. Informally, a family of functions is pseudorandom if no efficient algorithm can distinguish between a random function from the family and a truly random function. The existence of such families is one of the most basic assumptions in cryptography.

A complete account of natural proofs is available at many sources (such as [RR97] or [AB09]), and we don't aim to provide it in this survey. Our main goal is to define and discuss an analog of this barrier result for algebraic circuits (and, in doing so in Section 3, we will also recall the main properties of the definition of Razborov and Rudich [RR97]). We do like to stress two key points. The first is that, unlike other barriers, the natural proofs barrier is based on the (widely believed, but nevertheless an unproven) assumption that pseudorandom function families exist. Thus, when defining an algebraic analog, it's reasonable to base it on complexity theoretic assumptions, and discuss the veracity of these assumptions. The second key point is the powerful explanatory power of the natural proofs paradigm: in many senses, the natural proofs barrier explains exactly why progress in boolean circuit lower bounds has significantly slowed. It is precisely because circuit classes which are only slightly stronger than those we can prove lower bounds for, are already believed to be able to compute pseudorandom functions.

The theory of algebraic natural proofs we present in this paper regrettably lies on shakier assumptions than the boolean natural proofs paradigm, and lacks its powerful explanatory power. The most important problem we'll present in this survey is to try to base the barrier on firmer and more believable assumptions. Nevertheless, it turns out that the meta-mathematical concepts of arguing about proofs of lower bounds for algebraic circuit models also leads to insights about the circuit models themselves, and the notion of algebraic natural proofs is tightly connected to many other important questions arising in algebraic complexity theory, such as the derandomization of the polynomial identity testing problem, geometric complexity theory, and the complexity of the minimum circuit size problem. We describe some of these connections in Section 3.

We begin by presenting a basic background on algebraic complexity theory and defining the main computational model we'll discuss in this survey.

2 Algebraic Complexity

2.1 Algebraic Circuits

The basic objects of study in boolean computation are boolean functions. There are numerous computational models and complexity measures that attempt to capture the complexity of computing boolean functions with respect to various resources. Algebraic complexity theory, on the other hand, studies the resources required for computing algebraic problems using algebraic computational models, and the analogous basic objects of study are multivariate polynomials. Indeed,

common algebraic problems such as computing the determinant or the permanent of a matrix, multiplying two matrices, or computing the discrete Fourier transform, are all polynomial functions (or multi-output polynomial maps) in the input variables.

The most general model for algebraic computation is an *algebraic circuit*. An algebraic circuit over a field $\mathbb{F}$ is a directed, acyclic graph, whose input gates are labeled by inputs variables from $\{x_1, \ldots, x_n\}$ or constants from $\mathbb{F}$, and whose internal nodes are labeled by either $+$ or $\times$. The circuit computes a polynomial in a natural way. As usual, a circuit whose underlying graph is a tree is called a *formula*. The size of the circuit, which is roughly the analog of the running time of the algorithm, is defined to be the number of edges in the circuit.

We note that this model indeed abstracts almost all of the clever and efficient algorithms known for algebraic problems: the efficient algorithms for computing the determinant [Csa76, Ber84, MV99], multiplying matrices [Str69, Blä13] or computing the discrete Fourier transform [CT65] are all in fact constructions of small algebraic circuits.

As in boolean computation, one can define complexity classes for multivariate polynomials. The two most important ones are VP, the class of (families of) n-variate polynomials of degree $\mathsf{poly}(n)$ that are computed by $\mathsf{poly}(n)$-sized algebraic circuits; and VNP, which informally captures the class of all "explicit" polynomials.

A major open problem in algebraic complexity is to show that $\mathsf{VP} \neq \mathsf{VNP}$, i.e., to prove super-polynomial lower bounds on the size of algebraic circuits computing explicit polynomials. This is the algebraic analog of the P vs. NP problem. When computing a polynomial f using algebraic circuits, we insist on computing f syntactically, unlike boolean circuits which can compute a boolean function in many functionally equivalent ways (for example, the polynomial x^2 is not the same polynomial as x over $\mathbb{F}_2$, even though the function they compute is the same). Thus, algebraic circuits are more limited than boolean circuits, and VP vs. VNP is believed to be much easier than P vs. NP (indeed, in some formal sense the algebraic separation implies the boolean separation [Bür00]). Yet, there aren't any strong lower bounds known for general algebraic circuits.

Much like the state of affairs regarding boolean circuits, progress has been made regarding proving lower bounds for limited models of algebraic computation (we refer to [SY10, Sap16] for comprehensive surveys on lower bounds for algebraic computational models), but it remains unclear whether these methods can prove lower bounds for stronger models of computation. Further, it appears that all of these methods apply equally well to "easy" polynomials, i.e., polynomials that can be efficiently computed using small algebraic circuits, and it could be the case that this is intrinsic to the current methods.

2.2 Polynomial Identity Testing

Another important problem in algebraic complexity is the derandomization of the polynomial identity testing problem. This can be thought of as the algebraic analog of P vs. BPP. The input for the polynomial identity testing problem is an algebraic circuit C with n inputs, and the task at hand is to decide whether C computes the identically zero polynomial. The Schwartz-Zippel lemma implies an easy randomized algorithm for this problem: simply pick a random $\mathbf{a} \in S^n$ where $S \subseteq \mathbb{F}$ is a large enough subset of the field, and evaluate C at $\mathbf{a}$. Despite the simplicity of the randomized algorithm, there are no efficient deterministic algorithms for this problem. One explanation for this is a sequence of results on hardness vs. randomness in the algebraic setting that relates the identity testing problem to the challenge of proving algebraic circuit lower bounds (see [KS19] for a recent survey of the many available results).

The two models in which the problem is studied are the *white-box* model, in which the algorithm is given the circuit as an input and is allowed to inspect its structure, and the *black-box* model, in which the algorithm is only allowed to evaluate the circuit on inputs of its choice (note that the randomized algorithm above is a black-box algorithm). An efficient deterministic black-box algorithm is equivalent to constructing a small and explicit *hitting set*: a hitting set for a class of N-variate polynomials $\mathcal{C}$ is a set $\mathcal{H} \subseteq \mathbb{F}^N$ such that for any non-zero circuit C from the class $\mathcal{C}$, there is $\mathbf{a} \in \mathcal{H}$ such that $C(\mathbf{a}) \neq 0$. Standard counting arguments imply that (non explicit) small hitting sets do exist for "small" classes of polynomials, such as the class of polynomials with polynomial size arithmetic circuits.

A lot of progress has been made regarding identity testing for limited models of algebraic computation, both in the white-box and black-box models. For details on these results, we refer to the surveys [SY10, Sax09, Sax14].

3 Algebraic Natural Proofs

In several works ([AD08, Gro15]) it has been observed that almost all the lower bound techniques in algebraic complexity follow a certain strategy, which is in and of itself algebraic. As a leading example, consider so-called rank-based methods (which we revisit in Section 5), which include the partial derivative method and shifted partial derivative method. They work roughly as follows: given a polynomial f, one constructs a matrix M_f whose entries are determined by the coefficients of f (and thus typically, the dimension of M is exponential in n, the number of variables of f). Then, one shows that for any polynomial g computed by a small circuit from a certain class, $\text{rank}(M_g) < r$, whereas for some hard polynomial h, $\text{rank}(M_h) \geq r$. By well known properties of the determinant, this implies that the determinant of some $r \times r$ submatrix of M_h is non-zero, whereas the determinant of the same submatrix of M_g is zero, for every g with a small circuit.

In other words, what the lower bound proof shows is that, for the computational model at hand, there exists a non-zero N-variate polynomial P (the determinant of the $r \times r$ submatrix) whose variables correspond to the coefficients of n-variate degree-d polynomials, such that for all polynomials f that are easily computable (in that computational model), $P(\mathsf{coeff}(f)) = 0$, where $\mathsf{coeff}(f)$ denotes the vector of coefficients of f. We now observe that $\{f : P(\mathsf{coeff}(f)) = 0\}$ is a "natural" property in the sense of Razborov and Rudich [RR97].

1. Usefulness: by definition, if f has a small circuit then $P(\mathsf{coeff}(f)) = 0$.

2. Constructiveness: in the example above, P is a determinant of some matrix whose entries are determined by the coefficients of f. This matrix is exponentially large in the number of variables of f, but it is often only polynomially large in the number of coefficients of f. Thus, P can be efficiently computed when the complexity is measured as a function of the number of variables of P.

3. Largeness: unlike the case for boolean functions, the largeness property comes here "for free": since $P(\mathsf{coeff}(h)) \neq 0$, P is a non-zero polynomial, and thus $P(\mathsf{coeff}(g)) \neq 0$ for a "generic" polynomial g.

Following this example, we define the concept of *algebraic natural proofs*, which was introduced in the works of Forbes, Shpilka, and the author [FSV18] and Grochow, Kumar, Saks, and Saraf

[GKSS17]. These are proofs that use "algebraic distinguishers" (i.e., multivariate polynomials that are zero on coefficient vectors of "easy" polynomials) that are themselves efficiently computable.

We let $\mathbb{F}[x_1, \ldots, x_n]_{\leq d}$ denote the set of n-variate polynomials of degree at most d, and set $N = \binom{n+d}{n}$. For such a polynomial f, we let $\mathsf{coeff}(f) \in \mathbb{F}^N$ denote its vector of coefficients.

Definition 3.1 ([FSV18, GKSS17])**.** *Let $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]_{\leq d}$ be a class of polynomials, and let $\mathcal{D} \subseteq \mathbb{F}[X_1, \ldots, X_N]$. A non-zero polynomial $P \in \mathcal{D}$ is a $\mathcal{D}$-natural proof against $\mathcal{C}$, if for all $f \in \mathcal{C}$, $P(\mathsf{coeff}(f)) = 0$*

It is now natural to ask how strong are such proofs. Many of the existing lower bounds for algebraic models of computation indeed fall within this framework, and provide VP-natural properties (see, for example, [Gro15]). There are, however, some exceptions, as noted by [FSV18]. One could wonder how far these techniques can be pushed, which motivates the following question:

Question 3.2. *Let $\mathcal{D}$ to be the class of N-variate polynomials of degree at most $\mathsf{poly}(N)$ with algebraic circuit of size at most $\mathsf{poly}(N)$. Is there a $\mathcal{D}$-natural proof against the class of n-variate polynomials of degree $\mathsf{poly}(n)$ that can be computed by circuit of size $\mathsf{poly}(n)$?*

Question 3.2 is essentially the question of whether there are VP-natural proofs against the class VP, the algebraic analog of P. Having defined what are natural proofs, we turn our attention to studying the limitations of this framework, i.e., to the question of whether there's a natural proofs *barrier*.

3.1 Succinct Hitting Sets and the Algebraic Natural Proofs Barrier

Recall that the natural proofs barrier of Razborov and Rudich is a conditional statement, which depends on the existence of strong pseudorandom generators. Such generators are efficiently computable, but also "fool" all efficient distinguishers, which means that there's no efficient distinguishers that can tell apart random functions from efficiently computable functions.

Since the algebraic properties above concern vanishing of polynomials, translating this argument to the algebraic world results in a natural statement about *hitting sets* (which were defined in Section 2.2). That is, suppose there's a $\mathcal{D}$-natural proof against the class $\mathcal{C}$. Then, there's some non-zero $P \in \mathcal{D}$ which vanishes on $\mathcal{H} := \{\mathsf{coeff}(f) : f \in \mathcal{C}\}$. Equivalently, $\mathcal{H}$ is *not* a hitting set for P and consequently for $\mathcal{D}$. Taking the contrapositive, we state the following basic theorem, which immediately follows from the definitions.

Theorem 3.3 ([FSV18, GKSS17])**.** *Let $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]_{\leq d}$, and $\mathcal{D} \subseteq \mathbb{F}[X_1, \ldots, X_N]$ for $N = \binom{n+d}{d}$. If the set $\mathcal{H} := \{\mathsf{coeff}(f) : f \in \mathcal{C}\}$ is a hitting set for $\mathcal{D}$, then there are no $\mathcal{D}$-natural proofs against $\mathcal{C}$.*

The class $\mathcal{H} \subseteq \mathbb{F}^N$ above has the property that every element in the set, which is an N-dimensional vector, can be succinctly described using a circuit from $\mathcal{C}$, which is usually a much smaller description. For example, when $d = \mathsf{poly}(n)$ and $\mathcal{C} = \mathsf{VP}$, the description is of size $\mathsf{poly}(n)$ whereas N is exponential in n. We call such hitting sets *$\mathcal{C}$-succinct hitting sets*.

The main conjecture underlying the algebraic natural proofs barrier is that such succinct hitting sets exist.

Question 3.4. *Are there succinct hitting sets? In particular, for $N := \binom{n+d}{d}$, are the coefficients vectors of n-variate polynomials computable by circuits of size $\mathsf{poly}(n, d)$ a succinct hitting set against the class of N-variate polynomials of degree $\mathsf{poly}(N)$ computable by circuits of size $\mathsf{poly}(N)$?*

A priori, when the underlying field is infinite, a succinct hitting set doesn't even need to be finite, and the relation between this question and the standard setting of black box polynomial identity testing, in which one tries to construct small and explicit hitting sets, might not be clear. However, completeness results in algebraic complexity theory imply that if there's *any* succinct hitting sets, then there's an explicit succinct hitting set of quasipolynomial size.

Theorem 3.5 ([FSV18]). *Let $\mathbb{F}$ be a field. There's an explicit set $\mathcal{H} \subseteq \mathbb{F}^N$ of quasipolynomial size with the following property: if the answer to Question 3.4 is positive, then $\mathcal{H}$ is a hitting set for the class of N-variate polynomials of degree $\mathsf{poly}(N)$ computable by circuits of size $\mathsf{poly}(N)$.*

Thus, providing a positive answer to Question 3.4 would require, at the very least, derandomizing polynomial identity testing, and therefore this question might be difficult to resolve. In Section 4 we review some of the evidence towards positive and negative answers to this question.

3.2 Equations For Varieties and Geometric Complexity Theory

In this section we describe a different point of view regarding algebraic natural proofs, coming from algebraic geometry. It will be convenient to assume for this section that the underlying field $\mathbb{F}$ is algebraically closed.

Let $\mathcal{C}$ be an algebraic complexity class. By thinking of the (coefficient vectors of) elements in $\mathcal{C}$ as points in $\mathbb{F}^N$, we can think of $\mathcal{C}$ as a subset of $\mathbb{F}^N$.

A *variety* $V \subseteq \mathbb{F}^N$ is a set of solutions to polynomial equations. That is, V is a variety if there exist polynomials $f_1, \ldots, f_t \in \mathbb{F}[X_1, \ldots, X_N]$ such that $V = \{\mathbf{X} \in \mathbb{F}^N : f_1(\mathbf{X}) = f_2(\mathbf{X}) = \cdots = f_t(\mathbf{X}) = 0\}$ (such sets are more precisely called *affine varieties*, and are sometimes referred to as *algebraic sets*. For simplicity, we will use the name varieties). We also say that $f_1, \ldots, f_t$ *define* the variety V. Note that given a variety V, the set of defining polynomials in not unique, but Hilbert's basis theorem states that there's always a finite set of defining polynomials.

For a set $\mathcal{C} \subseteq \mathbb{F}^N$ we define the ideal $\mathbf{I}(\mathcal{C}) \subseteq \mathbb{F}[X_1, \ldots X_N]$ to be the set of polynomials that vanish on $\mathcal{C}$. We refer to a non-zero element $P \in \mathbf{I}(\mathcal{C})$ as an *equation* for $\mathcal{C}$.

Each variety V corresponds to an $\mathbf{I}(V)$, and conversely, each ideal $I \subseteq \mathbb{F}[X_1, \ldots, X_n]$ corresponds to a variety

$$\mathbf{V}(I) = \{\mathbf{X} : F(\mathbf{X}) = 0 \text{ for all } F \in I\}.$$

The (Zariski) *closure* of a set $\mathcal{C}$, denoted $\overline{\mathcal{C}}$, is the set $\mathbf{V}(\mathbf{I}(\mathcal{C}))$. In words, the closure of $\mathcal{C}$ is the set of common zeros of all the polynomials that vanish on $\mathcal{C}$. It is also the smallest variety with respect to inclusion which contains $\mathcal{C}$. By construction, $\overline{\mathcal{C}}$ is a variety, and a polynomial which vanishes everywhere on $\mathcal{C}$ also vanishes on $\overline{\mathcal{C}}$. We think of $\overline{\mathcal{C}}$ as the variety associated with the class $\mathcal{C}$.

Over the complex numbers, it is instructive to think of the Zariski closure of $\mathcal{C}$ as the usual Euclidean closure, since for the classes $\mathcal{C}$ we consider in this text, it can be shown that these two notions of closure coincide (see, e.g., Section 4.2 of [BI17]).

In this terminology, observe that an algebraic natural proof is a non-zero polynomial $P \in \mathbf{I}(\mathcal{C})$. But note that by showing that $P(\mathsf{coeff}(g)) \neq 0$, we actually prove the slightly stronger statement that $g \notin \overline{\mathcal{C}}$. That is, algebraic proofs always show a lower bound against the *closure* of the given class $\mathcal{C}$. This is an advantage, since the lower bound is stronger, but it might also be a disadvantage, since, if, for example, it turned out that $\overline{\mathsf{VP}} = \mathsf{VNP}$ (this is strongly suspected to be false), then by

definition algebraic proofs can't separate VP from VNP. Among other reasons, this fact motivates studying the closure of algebraic complexity classes, although this is not the focus of this survey.

In this formulation, proving lower bounds is an instance of the more general problem of *variety membership testing* [BIJL18, BIL$^+$19, BIL$^+$21]. This is the problem of, given a variety V and a point x, to decide whether $x \in V$. Of course, in order to specify this as an algorithmic problem, one needs to describe how is the variety V given as an input. For example, if we're given algebraic circuits for the polynomials $f_1, \ldots, f_t$ whose zero set defines V, then this problem is easy and is solved by simply evaluating the circuits on the point x. The common case in varieties related to algebraic complexity theory, however, is that these polynomials are not known (indeed, finding them is usually almost as hard as proving a lower bound!), and yet the variety does have a different short description using other means, e.g., as the closure of the image of an explicitly defined polynomial map (as a concrete example, for each s, the class of coefficient vectors of polynomials computed by a circuit of size at most s can be shown to be the image of an explicit polynomial map of degree $\mathsf{poly}(s)$ in $\mathsf{poly}(s)$ many variables).

Bläser, Ikenmeyer, Jindal, and Lysikov [BIJL18] and Bläser, Ikenmeyer, Lysikov, Pandey, and Schreyer [BIL$^+$19] show that some natural varieties (whose membership problem is NP-hard) must have equations which require circuits of super-polynomial size, unless $\mathsf{coNP} \subseteq \mathsf{MA}$, and in particular, the polynomial hierarchy collapses (which is a much more standard complexity assumption). As an example, we'll sketch an instance of this result for tensors with small *slice rank*.

An $n \times n \times n$ tensor is a trilinear polynomial T in $x_1, \ldots, x_n, y_1, \ldots, y_n, z_1, \ldots, z_n$, that is, $T = \sum_{i,j,k} a_{i,j,k} x_i y_j z_k$ (note that here the coefficients vector $(a_{i,j,k})$ has length n^3, which is polynomial in n, rather than exponential in n).

Definition 3.6. *A tensor T has slice rank 1 if there's a linear function $\ell(\cdot)$ and a bilinear polynomial $p(\cdot,\cdot)$ such that one of the following holds:*

1. $T = \ell(\mathbf{x}) \cdot p(\mathbf{y}, \mathbf{z})$
2. $T = \ell(\mathbf{y}) \cdot p(\mathbf{x}, \mathbf{z})$
3. $T = \ell(\mathbf{z}) \cdot p(\mathbf{x}, \mathbf{y})$

For a tensor T, its slice rank is the minimal r such that there exist r tensors of slice rank 1, $T_1, \ldots, T_r$, such that $T = T_1 + \cdots + T_r$.

We also say that T is of rank r and type (r_1, r_2, r_3) if $r_1 + r_2 + r_3 = r$ and

$$T = \sum_{p=1}^{r_1} T_p + \sum_{q=1}^{r_2} T'_q + \sum_{s=1}^{r_3} T''_s$$

where the T_p's are of the first type above, T'_q's are of the second type, and T''_s's are of the third type.

The slice rank of every polynomial is an integer $r \in [0, n]$. It's not hard to construct explicit tensors of high slice rank, so this may look like a toy setting for circuit lower bounds. However, we note that for the closely related problem of *tensor rank*, it's a major open problem to construct a three-dimensional $n \times n \times n$ tensor whose tensor rank is $\omega(n)$, and such a construction will imply new circuit lower bounds (we define and discuss this further in Section 5).

Going back to slice rank, we present the following two facts from [BIL$^+$19].

Fact 3.7. *For every r, the set SR_r of coefficient vectors of tensors whose slice rank is at most r is a variety.*

Fact 3.8. *The language $\{(T, r) : \text{the slice rank of } T \text{ is at most } r\}$ is* NP*-hard.*

The above two facts imply that for some r, the *non*-containment problem for the variety SR_r is coNP-hard. Intuitively, this may suggest that this variety must have equations that are hard to compute. However, note that each specific equation P can only be used to prove non-containment for some vectors (those for which $P(x) \neq 0$), but could potentially vanish on all the hard instances. Nevertheless, the methodology of [BIJL18, BIL+19] can be used to prove the following.

Theorem 3.9. *For some $r \in [0, n]$, every set of defining equations for SR_r contains an equation P which requires algebraic circuits of super-polynomial size, unless* coNP $\subseteq$ MA *and in particular the polynomial hierarchy collapses.*

We remark that a simple dimension argument can be used to show that there are equations for SR_r of degree poly(n) [KV21]. Thus, Theorem 3.9 doesn't follow trivially from showing that all equations have extremely high degree.

Proof Sketch. Suppose that $SR_r \subseteq \mathbb{F}^{n \times n \times n}$ can be defined as the set of common zeros of polynomials $P_1, \ldots, P_m$, all of them having polynomial circuit complexity. The main idea is to show that this assumption implies an MA algorithm for non-containment in SR_r, which is a coNP-hard problem.

Given v, the algorithm would expect the prover Merlin to supply a circuit for an equation P such that $P(v) \neq 0$. If indeed $v \notin V$, by assumption there exists such P with polynomial circuit complexity, so Merlin can indeed describe P in using a string of polynomial length[3], and evaluating $P(v)$ can also be done in polynomial time.

The only challenge is for the verifier to check that P is indeed an equation for SR_r. This is done as follows. For each (r_1, r_2, r_3) such that $r_1 + r_2 + r_3 = r$, we define a polynomial map $\Gamma_{r_1,r_2,r_3} : \mathbb{F}^m \to \mathbb{F}^{n \times n \times n}$ (where $m = (n^2+n)(r_1+r_2+r_3)$) of degree 2 and poly(n) size circuit, whose image is precisely all the coefficient vectors of tensors of rank r and type (r_1, r_2, r_3). Given such a family of maps, checking that P is indeed an equation amounts to verifying that the composed circuit $P(\Gamma_{r_1,r_2,r_3}(\mathbf{x}))$ is identically zero for all (r_1, r_2, r_3). This verification can be done by Arthur in randomized polynomial time, as this is simply an instance of polynomial identity testing.

It remains to describe the construction of Γ_{r_1,r_2,r_3}. The coefficient vector of a "generic" tensor of slice rank 1 of the form $\ell(\mathbf{x})p(\mathbf{y}, \mathbf{z}) = \sum_{i,j,k} u_i v_{j,k} x_i y_j z_k$ is given by the map $\Gamma^1 : \mathbb{F}^{n^2+n} \to \mathbb{F}^{n \times n \times n}$ defined as $\Gamma^1(\mathbf{u}, \mathbf{v})_{(i,j,k)} = u_i v_{j,k}$. Similarly one may define Γ^2 and Γ^3 for the other two types of tensors of slice rank 1 by merely changing the indexing. The final map Γ_{r_1,r_2,r_3} is defined as

$$(\Gamma_{r_1,r_2,r_3})_{(i,j,k)} = \sum_{p=1}^{r_1} \Gamma^1_p + \sum_{q=1}^{r_2} \Gamma^2_q + \sum_{s=1}^{r_3} \Gamma^3_s$$

where all the Γ^b_c's are defined over disjoint sets of variables. By definition, every tensor of type (r_1, r_2, r_3) is in the image of Γ_{r_1,r_2,r_3}. □

Refs. [BIJL18, BIL+19] give several other natural varieties for which this type of theorem holds.

[3]Strictly speaking, over infinite fields (such as $\mathbb{C}$), one has to also assume that the coefficients in the circuit are rationals of small enough size so that they have short bit complexity. We ignore this technical detail.

3.3 Algebraic Minimum Circuit Size Problem

We now describe a different point of view regarding algebraic natural proof, as the natural analog of the famous Minimum Circuit Size Problem (MCSP) from boolean complexity.

Recall that the Minimum Circuit Size Problem is the problem of, given as input the truth table of a boolean function $f : \{0,1\}^n \to \{0,1\}$ and a parameter s, to decide whether f can be computed by a circuit of size at most s. This problem turns out to be immensely important in the study of boolean computational classes and the relations between them, and it is an active object of research whose complexity is far from understood (see [All20] for a recent survey).

Under cryptographic assumptions, MCSP doesn't have a polynomial time algorithm. Nevertheless, researchers have struggled with classifying its complexity. For example, it's not known to be NP-hard, and a proof of it being NP-hard will imply new circuit lower bounds, and thus is might be difficult to prove that it's NP-hard [KC00, MW17]. It's also possible to show (unconditionally) that it's *not* NP-hard under some limited notions of reductions that suffice for many other NP-hardness results [MW17, All20]. Weak circuit lower bounds on MCSP also imply strong separations of complexity classes [OS18, CJW20].

A first attempt in defining an algebraic analog of MCSP for algebraic complexity is to consider the *boolean* problem of deciding, given as input the list of coefficients of a polynomial and a size parameter s, whether the input polynomial has an algebraic circuit of size at most s. That is, in this version the input is an algebraic object but the algorithm requested is an algorithm in the standard, boolean model (i.e., a Turing machine).

While it's possible to translate some of the results mentioned above to this setting, a big part of what makes MCSP so unique is its "meta complexity" nature, where the problem encodes the complexity of an object in the same model of the algorithm deciding the problem. Thus, in the algebraic case, it is more natural to require the algorithm as well to be a non-zero algebraic circuit which vanishes on (coefficient vectors of) polynomials whose circuit size is small.

This naturally leads us back to the natural proofs and equations for varieties formulations. Namely, pick s and d as some fixed functions of n (e.g., $s = d = n^{10}$). The complexity of the "algebraic MCSP" question with respect to n, d and s is asking what is the minimal size of a circuit for an N-variate polynomial which vanishes on all inputs which are coefficient vectors of polynomials with circuits of size at most s (as usual $N = \binom{n+d}{d}$).

Much like MCSP, it is interesting to understand the implications of "algebraic MCSP" being easy or hard (i.e., whether we should expect the circuit size to be polynomial in N or not).

Question 3.10. *What are the implications of the hardness or easiness of "algebraic MCSP"?*

We have already noted in Theorem 3.5 that (informally) the hardness of the algebraic MCSP problem implies derandomization of polynomial identity testing. Another partial answer to Question 3.10 was given by Kumar and the author [KV21] who used a technique similar to the one from Theorem 3.9 to show that in the setting of algebraic circuits computing linear transformations, if the corresponding version of algebraic MCSP is easy, then this implies an efficient construction with an NP oracle of linear transformations such that any algebraic circuit computing them must be of large size.

4 Is There an Algebraic Natural Proofs Barrier?

4.1 Succinct Hitting Sets for Restricted Models

As mentioned above, the barrier for the existence for algebraic natural proofs suggested by [FSV18, GKSS17] is the existence of succinct hitting sets. Proving their existence unconditionally will imply a derandomization of the polynomial identity testing and therefore also lower bounds, and is thus considered a difficult problem. However, there are restricted models of algebraic computation for which it is known how to derandomize the identity testing problem. Ref. [FSV18] shows that many of these constructions can be in fact made succinct.

The details of most constructions are somewhat technical, and the proofs for their correctness rely on previous works. Therefore, we only show in this survey a simple representative example: a succinct hitting sets for sparse polynomials.

A polynomial $F \in \mathbb{F}[X_1, \ldots, X_N]$ has sparsity s if F can be written as a sum of at most s monomials. This is a natural restricted class of polynomials for which many black box identity testing algorithms are known (see, e.g., [SY10, Sax09]).

In this section we'll show that there's a succinct hitting sets $\mathcal{H} \subseteq \mathbb{F}^N$ for this class. For simplicity, we'll assume $N = 2^n$ and we'll think of vector in $\mathbb{F}^N$ as representing coefficient vectors of *multilinear* n-variate polynomials. Each vector $\mathcal{H}$ will be a coefficient vector of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ computable by a depth 3 circuits of size $\mathsf{polylog}(s)$. In the natural case where $s = \mathsf{poly}(N)$ this size is polynomial in n.

To present this construction, we first start with the following lemma, showing how to hit polynomials that have a non-zero monomials of low support. The support of a monomial is the number of distinct variables appearing in the monomial with positive degree. For a vector $\mathbf{a} \in \mathbb{F}^N$ we denote by $\|\mathbf{a}\|_0$ its Hamming weight, i.e., the number of coordinates in which it's non-zero.

Lemma 4.1. *Let $F \in \mathbb{F}[X_1, \ldots, X_N]$ be a polynomial such that there exists a non-zero monomial in $\{X_1, \ldots, X_N\}$, with support of size at most k, whose coefficient in F is non-zero. Let*

$$\mathcal{H}_k = \left\{\mathbf{a} \in \mathbb{F}^N : \|\mathbf{a}\|_0 \leq k\right\}.$$

Then there's a vector $\mathbf{b} \in \mathcal{H}_k$ such that $F(\mathbf{b}) \neq 0$.

Proof. Let M be the corresponding monomial in F, and suppose without loss of generality that $M = X_1^{i_1} X_2^{i_2} \cdots X_k^{i_k}$. Observe that under the restriction $X_{k+1} = X_{k+2} = \cdots = X_n = 0$, F remains a non-zero polynomial (as the monomial M, which has a non-zero coefficient, survives this restriction, and can't be cancelled by any other monomial). Thus there exists a vector $\mathbf{b}$ whose last $n - k$ coordinates are zero (and in particular $\mathbf{b} \in \mathcal{H}_k$) such that $F(\mathbf{b}) \neq 0$. □

Every vector $\mathbf{b} \in \mathcal{H}_k$ is a coefficient vector of a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with at most k non-zero monomials, and thus f can be computed by a depth-2 circuit of size $\mathsf{poly}(k, n)$. For our setting we would want to set $k = \mathsf{polylog}(s)$, but it's clearly not true that every s-sparse polynomial $F \in \mathbb{F}[X_1, \ldots, X_N]$ has a non-zero monomial of support $\mathsf{polylog}(s)$. It turns out, however, that such a statement is true when considering $F(X_1 + 1, \ldots, X_N + 1)$.

Lemma 4.2 ([For15, GKST17, FSV18])**.** *Let $F \in \mathbb{F}[X_1, \ldots, X_N]$ be a polynomial with at most s monomials. Then the polynomial $F(X_1 + 1, \ldots, X_N + 1)$ has a monomial of support at most* $\log s$.

The observation that $\mathbf{1} := (1, \ldots, 1)$ is the coefficient vector of the multilinear polynomial $\prod_{i=1}^{n}(x_i + 1)$ suggests the following construction of a succinct hitting set for s-sparse polynomials.

Theorem 4.3. *Let $N = 2^n$. Let $\mathcal{C}$ be the class of n-variate polynomials computed by a depth 3 circuits of size $\mathsf{poly}(n, \log s)$. There's a $\mathcal{C}$-succinct hitting set for the class $\mathcal{D}$ of s-sparse N-variate polynomials.*

Proof. Consider the set $\mathcal{H} := \{\mathbf{1} + \mathbf{a} : \mathbf{a} \in \mathcal{H}_{\log s}\}$ (where $\mathcal{H}_{\log s}$ is as defined in Lemma 4.1). We first show $\mathcal{H}$ is a hitting set. Let $G = F(X_1 + 1, \ldots X_N + 1)$. By Lemmas 4.2 and 4.1, there's $\mathbf{b} \in \mathcal{H}_{\log s}$ such that $G(\mathbf{b}) \neq 0$, and therefore $F(\mathbf{1} + \mathbf{b}) = G(\mathbf{b}) \neq 0$.

Further, each element in $\mathcal{H}$ is the coefficient vector of a multilinear polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ of the form $\prod_{i=1}^{n}(x_i+1)+g(x_1, \ldots, x_n)$ where g has at most $\log s$ monomials. Thus, f is computable by a depth 3 circuit of size $\mathsf{poly}(n, \log s)$. □

Corollary 4.4. *Let $\mathcal{C}$ denote the class of n-variate multilinear polynomials with depth 3 circuits of size at most $\mathsf{poly}(n)$. Any equation for $\mathcal{C}$ must have sparsity which is super-polynomial in $N := 2^n$. Equivalently, letting $\mathcal{D}$ denote the class of N-variate polynomial with sparsity at most $\mathsf{poly}(N)$, there are no $\mathcal{D}$-natural proof against $\mathcal{C}$.*

Ref. [FSV18] contains many more examples of such succinct hitting sets for various models. Since they are somewhat technical in nature and rely extensively on previous work, we refer to [FSV18] for the details.

4.2 Equations with Small Circuits for Bounded Coefficients Polynomials

We now turn to a result that can be understood as an argument in favor of the existence of algebraic natural proofs for algebraic circuits (that is, the non-existence of an algebraic natural proofs barrier).

In Ref. [CKR+20], Chatterjee, Kumar, Ramya, Saptharishi, and Tengse prove the following:

Theorem 4.5 ([CKR+20])**.** *Let $c > 0$ be an arbitrary constant.*

1. *Let $\mathbb{F}$ be a fixed finite field of constant size. For $N = \binom{n+n^c}{n}$, there exists an N variate polynomial $P \in \mathbb{F}[X_1, \ldots, X_N]$ of degree $\mathsf{poly}(N)$, which can be computed by a circuit of size $\mathsf{poly}(N)$, such that for every n-variate polynomial f over $\mathbb{F}$ of degree at most n^c that can be computed by a circuit of size at most n^c, $P(\mathsf{coeff}(f)) = 0$. Further, there exists an n-variate polynomial h over $\mathbb{F}$ of degree at most n^c such that $P(\mathsf{coeff}(h)) \neq 0$.*
2. *For $N = \binom{n+n^c}{n}$, there exists an N variate polynomial $P \in \mathbb{Q}[X_1, \ldots, X_N]$, which can be computed by a circuit of size $\mathsf{poly}(N)$, such that for every n-variate polynomial f over $\mathbb{Q}$ of degree at most n^c that can be computed by a circuit of size at most n^c and has coefficients in the set $\{0, \pm 1\}$, $P(\mathsf{coeff}(f)) = 0$. Further, there exists an n-variate polynomial h over $\mathbb{Q}$ of degree at most n^c with coefficients in $\{0, \pm 1\}$ such that $P(\mathsf{coeff}(h)) \neq 0$.*

That is, Ref. [CKR+20] shows the existence of efficient "algebraic distinguishers" for circuit lower bounds for polynomials with small coefficients, which is an extremely interesting class of polynomials containing most interesting polynomials such as the permanent or determinant.

We note that unlike the case in the usual setting for algebraic natural proofs, the "further" part in both theorems does not follow automatically from P being a non-zero polynomial and requires

an extra argument. For example, the non-zero polynomial $X_1 \cdot (X_1 - 1) \cdot (X_1 + 1)$ is a non-zero polynomial which vanishes on all vectors in $\{0, \pm 1\}^N$. Further, these constructions are only able to show the existence of a non-zero h such that $P(\mathsf{coeff}(h)) \neq 0$, but it's unknown whether h is in VNP, and thus it's not clear yet which separations of complexity classes such constructions give. In fact, the proof shows the existence of a large set of such polynomials, but not large enough to be a significant enough fraction of the set of all polynomials from the relevant domain so as to satisfy the usual "largeness" condition for natural proofs. To summarize, this construction doesn't answer Question 3.4, but it could hint that, even if algebraic natural proofs do not exist, there are still "constructive" ways to prove lower bounds for interesting polynomials.

We'll sketch a proof of the first item of Theorem 4.5.

Proof Sketch of Theorem 4.5. Suppose for simplicity that $\mathbb{F} = \mathbb{F}_2$. Let $s = n^{\log n}$ (s can be chosen to be any super-polynomial function of f which grows slowly enough). Let $\mathbb{K}$ be an extension of $\mathbb{F}$ of size $O(n^{2c})$. Using counting arguments, it can be proved that there's a hitting set $\mathcal{H} \subseteq \mathbb{K}^n$ of size $O(s^2)$ for all n-variate polynomials of degree at most n^c that can be computed by a circuit of size at most s. In order to convey the main ideas of the proof, let's further assume that $\mathcal{H}$ is a subset of $\mathbb{F}^n$ and not merely $\mathbb{K}^n$ (we'll remark at the end how to handle the general case).

The polynomial P will be a product of 2 polynomials. The first is a polynomial computing the OR function over N variables, which is a degree N polynomial which can clearly be computed by a circuit of size $O(N)$. The second is a polynomial Q such that $Q(\mathbf{X}) = 0$ if and only if the vector $\mathbf{X}$ describes a coefficient vector of a polynomial which vanishes on $\mathcal{H}$. Thus, by assumption on $\mathcal{H}$, the non-zeros of P are coefficient vectors of non-zero polynomials that don't vanish on $\mathcal{H}$, i.e., have circuit size at least s.

We construct Q as follows. $Q = 1 - \prod_{\mathbf{a} \in \mathcal{H}} (1 - Q_{\mathbf{a}})$, where $Q_{\mathbf{a}}(\mathbf{X})$ is a polynomial which evaluates to 0 if and only if $\mathbf{X}$ describes the coefficient vector of a polynomial that evaluates to 0 at the point $\mathbf{a}$. This can be checked by a polynomial of degree 1, by simply taking the inner product between the input variables and the evaluation vector of $\mathbf{a}$ on all monomials on n variables of degree at most n^c, of which there are exactly N. The circuit complexity of Q is $O(N \cdot |\mathcal{H}|)$.

Finally, we'd like to show that P is not identically zero on $\mathbb{F}^N$. Note that each constraint of the form $g(\mathbf{a}) = 0$ for $\mathbf{a} \in \mathcal{H}$ is a homogeneous linear constraint on the coefficient of g. Since $|\mathcal{H}| \ll N$, this linear system has a non-zero solution, that is, there exists a non-zero h such that h vanishes on $\mathcal{H}$. Such h would satisfy $P(\mathsf{coeff}(h)) \neq 0$.

As mentioned in the beginning of the proof, we can't really assume that $\mathcal{H} \subseteq \mathbb{F}^n$ but rather we have to use the fact that $\mathcal{H} \subseteq \mathbb{K}^n$. This is done in a straightforward manner by treating elements of $\mathbb{K}$ as vectors over $\mathbb{F}$ of length $O([\mathbb{K} : \mathbb{F}]) = O(\log n)$ and modifying P accordingly. This incurs a small increase in the degree of P and circuit size of P, but this increase is rather meaningless with respect to the relevant parameters. □

4.3 A Natural Proofs Barrier for VNP

As described earlier, we would have liked to establish the natural proofs barrier on firmer grounds—ideally, on the most basic assumption in algebraic complexity theory, that $\mathsf{VP} \neq \mathsf{VNP}$. In Ref. [KRST20], Kumar, Ramya, Saptharishi, and Tengse obtain a related result, but for the class VNP itself: that is, they prove that if the Permanent polynomial requires circuits of size 2^{n^ε} for some fixed $\varepsilon > 0$, then any polynomial family P that vanishes on all inputs of the form $\mathsf{coeff}(g)$ for $g \in \mathsf{VNP}$ must have super-polynomial circuit size.

The proof of Kumar, Ramya, Saptharishi, and Tengse uses the equivalence between natural proofs and succinct hitting sets as in Section 3.1. A well known hardness vs. randomness theorem of Kabanets and Impagliazzo [KI04] (based on ideas due to Nisan and Wigderson [NW94] in the context of boolean derandomization) gives an explicit conditional construction of a hitting set for VP (under the assumption that the permanent requires exponential size algebraic circuits). In Ref. [KRST20], it is shown that this construction can be made VNP-succinct, that is, each element of the hitting set is a coefficient vector of a polynomial in VNP. Using the equivalence between natural proofs and succinct hitting sets, the conclusion now directly follows.

While VNP-succinctness is presumably much weaker then VP-succinctness, this result has several interesting corollaries. The first is a separation of the bounded coefficient setting, which we have dealt with in Section 4.2, from the general setting. While we have sketched a proof of Theorem 4.5 for polynomials in VP, an analogous theorem can be generalized to hold for polynomials with bounded coefficients from VNP without much extra work. That is, there's an efficiently computable polynomial which vanishes on all coefficient vectors of polynomials in VNP with bounded integer coefficients. However, the result of Ref. [KRST20] shows that it's unlikely that this continues to hold when removing the condition about bounded integer coefficients.

A second interesting corollary is the observation that, assuming that the permanent requires exponential size circuits, and further that natural proofs for VP *do* exist, then *any* such natural proof would automatically separate VP from VNP.

5 An Unconditional Barrier for Rank Based Techniques

We now describe a result of a different flavor, due to Efremenko, Garg, Oliveira, and Wigderson [EGOW18], which shows an *unconditional* barrier for proving lower bounds for a more limited class of proof methods which they call *rank based methods*. These methods, mentioned in Section 3 are more limited than general algebraic natural proofs, but are still quite general and common techniques for proving lower bounds for algebraic circuits.

Before defining precisely rank based methods, we'll define one model for which the barrier applies. Recall that a 3 dimensional $n \times n \times n$ tensor is a trilinear polynomial of the form $T(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{i,j,k=1}^{n} a_{i,j,k} x_i y_j z_k$. The *tensor rank* of a tensor T is the minimal r such that

$$T = \sum_{i=1}^{r} \ell_{i,1}(\mathbf{x}) \cdot \ell_{i,2}(\mathbf{y}) \cdot \ell_{i,3}(\mathbf{z})$$

where for every $i \in [r]$ and $j \in [3]$, $\ell_{i,j}$ is a linear function. It is easy to show, using counting arguments, that most 3-dimensional $n \times n \times n$ tensors have rank $\Omega(n^2)$, and yet, it is an open problem to show an explicit tensor whose rank is super-linear in n. It is also not hard to show (see [SY10, Sap16]) that a lower bound of s on the tensor rank of T implies a lower bound of s on any algebraic circuit computing T, and thus this is a very natural and interesting complexity measure.

A *rank based* method for lower bounds on tensor rank is a linear map $L : \mathbb{F}^{n^3} \to \mathbb{F}^{m \times m}$ such that for every rank one tensor T_0, $\mathrm{rank}(L(\mathsf{coeff}(T_0))) \leq r$, whereas for some tensor T, $\mathrm{rank}(L(\mathsf{coeff}(T))) \geq C$. Since L is a linear map and the rank function is subadditive, such a map L will imply a lower bound of C/r on the tensor rank of T.

Rank methods were indeed successful in proving lower bounds for many restricted models of algebraic computation. However, as Ref. [EGOW18] shows, such methods are incapable of proving

super-linear lower bounds on the rank of three-dimensional tensors (the result of [EGOW18] is more general and applies to d-dimensional tensors and for other models such as Waring rank, but for the sake of simplicity, we only present this special case in this survey). A *barrier* for such a method would show, in the above notation, that for any linear map L, if $\text{rank}(L(\mathsf{coeff}(T_0))) \leq r$ for every rank one tensor T_0, then for any tensor T, $\text{rank}(L(\mathsf{coeff}(T))) \leq O(nr)$. This implies that the method can't prove a lower bound better than $\Omega(n)$. Such a statement is shown in [EGOW18].

Theorem 5.1 ([EGOW18])**.** *No rank based method is capable of proving a super-linear tensor rank lower bound for three dimensional tensors.*

The proof of Theorem 5.1 uses the following lemma, which relates the rank of a "symbolic" matrix of polynomials $M(\mathbf{w})$ over the field $\mathbb{F}(\mathbf{w})$ to its rank of evaluations $M(\mathbf{a})$ over $\mathbb{F}$. Using the characterization of rank as the largest submatrix whose determinant is non-zero, the lemma is an immediate corollary of the Schwartz-Zippel lemma.

Lemma 5.2. *Let $M(\mathbf{w})$ be a $k \times m$ matrix of n-variates polynomials in variables $\mathbf{w}$ over an infinite field $\mathbb{F}$. Then*

$$\text{rank}_{\mathbb{F}(\mathbf{w})} M(\mathbf{w}) = \max\{\text{rank}_{\mathbb{F}} M(\mathbf{a}) : \mathbf{a} \in \mathbb{F}^n\}.$$

Proof Sketch of Theorem 5.1. We wish to show that for any linear map $L : \mathbb{F}^{n^3} \to \mathbb{F}^{m \times m}$, if for every rank 1 tensor $T_0 := \ell_1(\mathbf{x}) \cdot \ell_2(\mathbf{y}) \cdot \ell_3(\mathbf{z})$ it holds that $\text{rank}(L(\mathsf{coeff}(T_0))) \leq r$, then for every tensor T, $\text{rank}(L(\mathsf{coeff}(T))) \leq O(nr)$.

We'll once again take the approach of thinking about the coefficients of T_0 as symbolic variables and consider the map L as a polynomial map in those variables. That is, suppose $T_0 = (\sum_{j=1}^n c_{1,j}x_i) \cdot (\sum_{j=1}^n c_{2,j}y_j) \cdot (\sum_{j=1}^n c_{3,j}z_j)$ is a "generic" rank 1 tensor, where $\mathbf{c}_t = (c_{t,1}, \ldots, c_{t,n})$ for $t \in [3]$ are thought of as vectors of indeterminates.

We can write

$$L(\mathsf{coeff}(T_0)) = \sum_{i,j,k \in [n]} A_{i,j,k} \cdot c_{1,i}c_{2,j}c_{3,k}, \tag{1}$$

that is, $L(\mathsf{coeff}(T_0))$ is a polynomial in $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$ with matrix coefficients: $A_{i,j,k}$ are $m \times m$ matrices over $\mathbb{F}$. Note that this polynomial itself is trilinear in $\mathbf{c} = (\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$. $L(\mathsf{coeff}(T_0))$ can also be thought of as a matrix $M(\mathbf{c})$ of polynomials in variables $\mathbf{c}$. By assumption, for every $\mathbf{a} \in \mathbb{F}^n$, $\text{rank}_{\mathbb{F}}(M(\mathbf{a})) \leq r$, which implies by Lemma 5.2 that $\text{rank}_{\mathbb{F}(\mathbf{c})} M(\mathbf{c}) \leq r$. Therefore,

$$M(\mathbf{c}) = \sum_{i=1}^{r} \mathbf{f}_i \mathbf{g}_i^T$$

for vectors $\mathbf{f}_i, \mathbf{g}_i \in (\mathbb{F}(\mathbf{c}))^m$. That is, the entries of $\mathbf{f}_i, \mathbf{g}_i$ are in the field of rational functions in the variables $\mathbf{c}$ over $\mathbb{F}$. Since the entries of $M(\mathbf{c})$ are trilinear forms in $(\mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3)$, by standard homogenization and multilinearization techniques we can ensure that

$$M(\mathbf{c}) = \sum_{i=1}^{R} \tilde{\mathbf{f}}_i \tilde{\mathbf{g}}_i^T \tag{2}$$

for $R = O(r)$, where now $\tilde{\mathbf{f}}_i, \tilde{\mathbf{g}}_i$ are vectors of *polynomials* that "respect" the trilinear structure of $M(\mathbf{c})$: namely, for every $i \in [R]$ there is some $j \in [3]$ such either $\tilde{\mathbf{f}}_i$ is linear in $\mathbf{c}_j$ and $\tilde{\mathbf{g}}_i$ is bilinear

in the two other sets of variables $\{\mathbf{c}_k : k \in [3] \setminus \{j\}\}$, or $\tilde{\mathbf{g}}_i$ is linear in $\mathbf{c}_j$ and $\tilde{\mathbf{f}}_i$ is bilinear in the two other sets of variables.

Recall that an $m \times k$ matrix of polynomials $B(\mathbf{c})$ can also be thought of as a polynomial in $\mathbf{c}$ with coefficients from $\mathbb{F}^{m \times k}$. We'll denote by $\mathcal{C}(B)$ the subspace of $\mathbb{F}^{m \times k}$ spanned by these coefficients. For example, $\mathcal{C}(M)$ is the span of the matrices $A_{i,j,k}$ appearing in (1).

Since a linear function has n coefficients, it can be shown that for every $B \in \mathcal{C}(\tilde{\mathbf{f}}_i \tilde{\mathbf{g}}_i^T)$, where $\tilde{\mathbf{f}}_i, \tilde{\mathbf{g}}_i$ are as in (2), $\mathrm{rank}(B) \leq n$. This implies by subadditivity that for every $B \in \mathcal{C}(M(\mathbf{c}))$, $\mathrm{rank}(B) \leq nR = O(nr)$.

We'll finish by showing that for every tensor T it holds that $L(\mathsf{coeff}(T)) \in \mathcal{C}(M(\mathbf{c}))$. Indeed, this is true for rank one tensors as

$$L\left(\mathsf{coeff}\left(\left(\sum_{j=1}^{n} a_{1,j}x_j\right) \cdot \left(\sum_{j=1}^{n} a_{2,j}y_j\right) \cdot \left(\sum_{j=1}^{n} a_{3,j}z_j\right)\right)\right) = M(\mathbf{a}),$$

and thus also true for T, because T can be written as a sum of rank 1 tensors, and L is a linear function. □

6 Summary and Discussion

In thus survey, we presented the concept of algebraic natural proofs and discussed its many facets. We hope to have convinced the reader that (like many other similar cases in complexity theory), what started as a meta-mathematical quest to understand our failure in proving strong lower bounds for algebraic models of computation, turned out to be relevant to studying the model *itself* and to provide insight on algebraic computation.

We finish by stressing the two problems which we see as most important in this area. The first is to provide evidence towards resolving Question 3.2, i.e., basing the algebraic natural proofs barriers on standard complexity theoretic or cryptographic assumptions. The second is the more open-ended Question 3.10, which asks what are the complexity theoretic implications of either the existence or non-existence of the algebraic natural proofs barrier.

Acknowledgements We thank Rohit Gurjar, Mrinal Kumar, Shir Peleg, and Amir Shpilka for providing useful comments on earlier versions of this text.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[AD08] Scott Aaronson and Andrew Drucker. Arithmetic natural proofs theory is sought. Blog post, `http://www.scottaaronson.com/blog/?p=336`, 2008.

[All20] Eric Allender. The new complexity landscape around circuit minimization. In *Language and Automata Theory and Applications - 14th International Conference, LATA 2020, Milan, Italy, March 4-6, 2020, Proceedings*, volume 12038 of *Lecture Notes in Computer Science*, pages 3–16. Springer, 2020.

[AW09] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory*, 1(1):2:1–2:54, 2009.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150, 1984.

[BGS75] Theodore P. Baker, John Gill, and Robert Solovay. Relativizations of the P =? NP question. *SIAM J. Comput.*, 4(4):431–442, 1975.

[BI17] Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory. Lecture notes, `http://pcwww.liv.ac.uk/~iken/teaching_sb/summer17/introtogct/gct.pdf`, 2017.

[BIJL18] Markus Bläser, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. Generalized matrix completion and algebraic natural proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1193–1206. ACM, 2018.

[BIL+19] Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. Variety membership testing, algebraic natural proofs, and geometric complexity theory. *CoRR*, abs/1911.02534, 2019.

[BIL+21] Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. On the orbit closure containment problem and slice rank of tensors. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2565–2584. SIAM, 2021.

[Blä13] Markus Bläser. *Fast Matrix Multiplication*. Number 5 in Graduate Surveys. Theory of Computing Library, 2013.

[Bür00] Peter Bürgisser. Cook's versus Valiant's hypothesis. *Theor. Comput. Sci.*, 235(1):71–88, 2000.

[CJW20] Lijie Chen, Ce Jin, and R. Ryan Williams. Sharp threshold results for computational complexity. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 1335–1348. ACM, 2020.

[CKR+20] Prerona Chatterjee, Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the existence of algebraically natural proofs. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 870–880. IEEE, 2020.

[Csa76] László Csanky. Fast parallel matrix inversion algorithms. *SIAM J. Comput.*, 5(4):618–623, 1976.

[CT65] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comp.*, 19:297–301, 1965.

[EGOW18] Klim Efremenko, Ankit Garg, Rafael Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. In *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[For15] Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 451–465. IEEE Computer Society, 2015.

[FSV18] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory Comput.*, 14(1):1–45, 2018.

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017.

[GKST17] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complex.*, 26(4):835–880, 2017.

[Gro15] Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Comput. Complex.*, 24(2):393–475, 2015.

[KC00] Valentine Kabanets and Jin-Yi Cai. Circuit minimization problem. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA*, pages 73–79. ACM, 2000.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complex.*, 13(1-2):1–46, 2004.

[KRST20] Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. If VNP is hard, then so are equations for it. *CoRR*, abs/2012.07056, 2020.

[KS19] Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bull. EATCS*, 129, 2019.

[KV21] Mrinal Kumar and Ben Lee Volk. A polynomial degree bound on equations for non-rigid matrices and small linear circuits. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 9:1–9:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[MV99] Meena Mahajan and V. Vinay. Determinant: Old algorithms, new insights. *SIAM J. Discret. Math.*, 12(4):474–490, 1999.

[MW17] Cody D. Murray and R. Ryan Williams. On the (non) NP-hardness of computing circuit complexity. *Theory Comput.*, 13(1):1–22, 2017.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[OS18] Igor Carboni Oliveira and Rahul Santhanam. Hardness magnification for natural problems. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 65–76. IEEE Computer Society, 2018.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.

[Sap16] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, `https://github.com/dasarpmar/lowerbounds-survey/`, 2016.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

[Sax14] Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in computational complexity*, volume 26 of *Progr. Comput. Sci. Appl. Logic*, pages 131–146. Birkhäuser/Springer, Cham, 2014.

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13(3):354–356, 1969.

[SY10] 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.