The Computational Complexity Column

by

V. Arvind

Institute of Mathematical Sciences, CIT Campus, Taramani
Chennai 600113, India
arvind@imsc.res.in
http://www.imsc.res.in/~arvind

The area of Geometric Complexity Theory, initiated by Ketan Mulmuley and his collaborators in the late 1990's, is an ambitious research project exploring connections between algebraic geometry and circuit complexity with the aim of proving superpolynomial arithmetic circuit lower bounds. This perspective on complexity has been growing rapidly in recent years. In this interesting article, Josh Grochow surveys complexity theory questions on polynomial ideals, which is a fundamental object of study in algebra. He shows connections with circuit complexity and algebraic proof systems and discusses a number of open problems.

Complexity in ideals of polynomials: questions on algebraic complexity of circuits and proofs

Joshua A. Grochow*

Abstract

Given ideals $I_n \subseteq \mathbb{F}[x_1, \ldots x_n]$ for each n, what can we say about the circuit complexity of polynomial families f_n in those ideals, that is, such that $f_n \in I_n$ for all n? Such ideals and their cosets arise naturally in algebraic circuit lower bounds, geometric complexity theory, and algebraic proof complexity. For ideals generated by a single element, this is the question of relating the complexity of a polynomial to the complexity of its factors, which has a long and rich history. For general ideals, essentially nothing beyond that is known, even for ideals generated by just 2 elements. For a few examples of specific ideals of interest coming from circuit lower bounds or proof complexity, some lower bounds on polynomials in these ideals are known using succinct hitting sets (Forbes–Shpilka–Volk, *Theory Comput.*, 2019) and circuit techniques (Forbes–Shpilka–Tzameret–Wigderson, CCC 2016). In this survey, we review these connections & motivations, and raise many questions that we hope will help shed light on the complexity landscape of polynomials in ideals.

1 Introduction

Recent progress in algebraic circuit complexity and algebraic proof complexity is starting to reveal that ideals of polynomials (and their cosets) are key objects "behind the scenes" of many of the central questions in algebraic computational complexity and proof complexity. My goal here is to motivate these objects and their importance, to highlight our extreme

*Departments of Computer Science and Mathematics University of Colorado at Boulder. Email: jgrochow@colorado.edu.

lack of understanding of the complexity properties of ideals of polynomials, and to put forward some open questions that I hope are fruitful for building up our understanding of their structure.

As I'm hoping this column will serve not only seasoned researchers but also beginning graduate students or even advanced undergraduates, the main background I'll assume is some knowledge of polynomials, such as unique factorization. (Some standard knowledge of Boolean complexity theory will help with motivation, but is not strictly speaking necessary.) Toward this end, to make sure we're all on the same page, we start with the definitions.

Definition of rings and ideals. A *ring* R is a set (also denoted R) endowed with two operations $+, \cdot \colon R \times R \to R$, called "addition" and "multiplication" satisfying the axioms you might expect form ordinary numerical addition and multiplication: both operations are associative, addition is commutative, every element has a negative, there is an additive identity called $0 \in R$, there is a multiplicative identity $1 \in R$, and multiplication distributes over addition. Examples include $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, finite fields $\mathbb{F}_q$, polynomial rings $R[x_1, \ldots, x_n]$ (where R is another ring), and matrix rings $M_n(R)$. If multiplication is commutative, we call R a commutative ring. Unless otherwise specified, whenever we say "ring" we will mean "commutative ring."

An *ideal* is a subset $I \subseteq R$ that is (1) closed under addition and negation, and (2) closed under multiplication by arbitrary elements of R: if $i \in I$ and $r \in R$, then $ri \in I$. An ideal I is *generated* by a set $S = \{f_1, f_2, \ldots, \}$ if it is the smallest ideal containing S. In this case we write $I = \langle S \rangle = \langle f_1, f_2, \ldots \rangle$. It follows that $I = \{\sum_{i=1}^{k} f_i g_i : f_i \in S, g_i \in R, k \in \mathbb{N}\}$. Examples include the prime ideal $\langle p \rangle \subseteq \mathbb{Z}$ or $\langle x^2, y \rangle \subseteq \mathbb{F}[x, y]$. In general rings ideals need not be generated by any finite set, but in the rings we consider they will always be. A *coset* of I is $r + I = \{r + i : i \in I\}$ for some $r \in R$. The cosets of I in R form the quotient ring R/I, where $(r + I) + (r' + I) = (r + r') + I$ and $(r + I)(r' + I) = rr' + I$.

Complexity in ideals of polynomials. Now to the topic at hand. Let $R_n = \mathbb{F}[x_1, \ldots, x_{\nu(n)}]$ be a polynomial ring for each n and let $I_n \subseteq R_n$ an ideal for each n. We refer to such a sequence as a *family*, and when we want to refer to the family as a whole we may write $R = (R_n) = (R_n)_{n=1,2,3,\ldots}$, and similarly $I = (I_n)$. A family of elements $f = (f_n)_{n=1,2,3,\ldots}$ with $f_n \in R_n$ is said to be in I if $f_n \in I_n$ for each n.

Main (meta-)question. Given a family of ideals $I = (I_n)$—or

more generally cosets $(c_n + I_n)$—what can be said about the complexity of polynomial families $(f_n) \in (c_n + I_n)$?

In particular, there are many situations where we have a family of ideal cosets $c + I = (c_n + I_n)$, and we would like to prove complexity lower bounds on *all* families $f = (f_n)$ in $c + I$ (that is, $f_n \in c_n + I_n$ for all n). In Sections 4–6 we will outline some of these situations, and use them to motivate our questions about polynomials in ideals. Proving a lower bound on *all* families in an ideal is different in a crucial way from the usual setting of lower bounds: if we wanted to show a separation of complexity classes $\mathcal{C} \neq \mathcal{D}$, it suffices to find *one single* family f such that $f \in \mathcal{D}$ but $f \notin \mathcal{C}$. But in our setting we want to show that *every* family $f \in c + I$ is not in some complexity class $\mathcal{C}$. As hard as circuit lower bounds are to begin with, this is *a priori* a much harder task, so it seems worth developing some theoretical scaffolding to learn more generally whatever we can about structural complexity in (cosets of) ideals of polynomials. But first, some preliminaries on algebraic complexity.

Acknowledgments. This paper is roughly based on a talk given by the author at the Oxford / Clay Mathematics Institute (CMI) Workshop on Complexity Theory in 2018. Thanks to the organizers E. Allender, B. Green, and R. Santhanam for the invitation and support. Thanks to Peter Bürgisser and Mrinal Kumar for providing useful references on factoring polynomials. The author was partially supported by NSF grant DMS-1750319 during the preparation of this article.

2 Algebraic complexity: why and what

In algebraic complexity we are primarily concerned with questions such as the minimum number of algebraic operations $(+, \times, -, /)$ needed to compute a given polynomial. On the one hand, this is well-motivated in its own right: there are many numerical and algebraic algorithms which deal with numbers "on their own terms," only using the basic arithmetic operations, and never opening up the machine representation in terms of bits. On the other hand, it is now well-known (but worth repeating) that algebraic complexity theory has close relationships with Boolean complexity theory. Before coming to these relationships, we begin with the definitions.

2.1 Definitions in algebraic complexity

We say a family $f = (f_n)$ is *p-bounded* if the number of variables and the degree of f_n are at most $\mathrm{poly}(n)$. We will mostly be concerned with p-bounded families.

An algebraic circuit over a field $\mathbb{F}$ is a directed acyclic graph, whose sources are labeled by (not necessarily distinct) variables x_i or constants from $\mathbb{F}$, whose internal nodes are labeled either $+$ or $\times$, and whose sinks are the output nodes. Each node computes a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ in the natural manner. The *size* of a circuit is its total number of nodes (including input nodes—while this is not exactly standard, it yields the same results except for very low-level complexity classes, and it will simplify some of our statements), and its *depth* is the largest distance between any source and any sink. Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, its *circuit size* is the minimum size of any circuit computing f, denoted $C(f)$. Here, even if $\mathbb{F}$ is a finite field, we mean computing f *symbolically* as a polynomial. For example, if we consider the polynomial $f(x) = x^p - x$, this evaluates to zero at all elements of $\mathbb{F}_p$, but even if we were working over $\mathbb{F}_p$ we would *not* say that the 0 circuit computes f; the circuit size of f is essentially $\log_2 p$, by repeated squaring. A family of polynomials $f = (f_n)$ is in Valiant's class VP if it is p-bounded and $C(f_n) \leq \mathrm{poly}(n)$. VP is a (nonuniform) algebraic analogue of P, or more accurately, NC^2, because of the result that any $f \in \mathsf{VP}$ can be computed by a poly-size family of circuits whose depth is at most $O(\log^2 n)$ [41].

To relate polynomial families to one another, we use standard notions of reduction. A *projection* over a field $\mathbb{F}$ is a map from a set of variables $\{x_1, \ldots, x_n\}$ to another set of variables and constants $\{y_1, \ldots, y_m\} \cup \mathbb{F}$. A polynomial $f(\vec{y})$ is a *projection* of a polynomial $g(\vec{x})$ if there is a projection $\pi \colon \{x_1, \ldots, x_n\} \to \{y_1, \ldots, y_m\} \cup \mathbb{F}$ such that $f(\vec{y}) = g(\pi(\vec{x}))$. A polynomial family (f_n) is a *p-projection* of another family (g_n) if there is a polynomial $r(n)$ such that f_n is a projection of $g_{r(n)}$ for all n, and in this case we write $f \leq_p g$. Most reductions in the literature are p-projections. Two polynomial families f, g are said to be of the same p-degree if $f \leq_p g$ and $g \leq_p f$, in which case we write $f \equiv_p g$. The p-degree of f is the set of all polynomial families of the same p-degree as f.

A slightly more general notion of reduction—the algebraic analogue of polynomial-time Turing reductions—is that of circuit reductions. Given a polynomial family f, we use $C^f(g)$ to denote the minimum size circuit computing g using arithmetic gates as well as f gates. For two polynomial families $f = (f_n), g = (g_n)$, we say that g *c-reduces* to f, denoted $g \leq_c f$, if $C^f(g_n) \leq \mathrm{poly}(n)$. Two polynomial families f, g are c-equivalent, denoted

$f \equiv_c g$, if $f \leq_c g$ and $g \leq_c f$; the set of all polynomials c-equivalent to g is its *c-degree*. VP is closed under both p-projections and c-reductions.

A family of polynomials $f = (f_n)$ is in Valiant's class VNP if it there is another family $g \in \mathsf{VP}$ and a polynomial $r(n)$ such that

$$f_n(\vec{x}) = \sum_{\vec{y} \in \{0,1\}^m} g_{r(n)}(\vec{x}, \vec{y}).$$

VNP is closed under p-projections. The permanent is VNP-complete under p-projections, and computing the permanent of $\{0,1\}$-matrices is #P-complete, so VNP is often thought of as the algebraic analogue of NP or, a bit more accurately, #P.

2.2 Connections between algebraic and Boolean complexity

Now we come to some of the relationships between the Boolean and algebraic worlds. First, Bürgisser showed [3] that certain algebraic separations are consequences of Boolean ones, so the algebraic separations are natural and necessary first targets. In particular, $\mathsf{P/poly} \neq \mathsf{NP/poly}$ (or even $\mathsf{NC}^3\mathsf{/poly} \neq \mathsf{PH/poly}$) implies $\mathsf{VP}_{\mathbb{F}} \neq \mathsf{VNP}_{\mathbb{F}}$ over all finite fields $\mathbb{F}_{p^k}$ and, assuming the Generalized Riemann Hypothesis, over all fields $\mathbb{F}$ of characteristic zero as well.

Second, Valiant [40] argued that "linear algebra offers essentially the only fast technique for computing multivariate polynomials of moderate degree," suggesting that the algebraic question of whether the permanent could be computed efficiently by means of linear algebra (a moderate-sized determinant) gets at the heart, or at least a key part of, the P versus NP question.

Third, a celebrated line of research initiated by Kabanets and Impagliazzo [24], then continued by Jansen & Santhanam [21] and Carmosino, Impagliazzo, Kabanets, and Kolokolova [7] says that if testing whether an algebraic circuit computes the identically zero function can be done in P (or even slightly sub-exponential deterministic time) rather than the current upper bound of coRP, then one gets lower bounds on algebraic circuit size.

Taken together, these results are strong motivation for studying algebraic complexity as a way of getting at key issues in Boolean complexity.

3 What is known about complexity in ideals

In terms of complexity of ideals in general, essentially the only known results are about *principal* ideals: those generated by a single element. For certain specific non-principal ideals of interest, more is known, which we cover in Sections 4.1 and 5, but for general ideals we do not know of any other results. The results on principal ideals were all originally phrased in terms of factors, viz. pairs of polynomials f, P such that f divides P. We will try to stick to this notation as much as possible: "f" for "factor" and P for "polynomial." In terms of ideals, this is the same as saying that P lies in the principal ideal $\langle f \rangle$.

These results are stated for fields of characteristic zero; they should all still work for fields of sufficiently large characteristic.

Theorem 3.1 (Kaltofen [22], Bürgisser [1, Thm. 8.14]; see also [9] for a new proof)**.** *Over any field of characteristic zero, suppose $P = f^e g$ with f, g coprime polynomials. Then*

$$C(f) \leq \mathrm{poly}(C(P), \deg f, e)$$

In fact, if $P_n = f_n^{e_n} g_n$ for all n and $e_n \leq \mathrm{poly}(n)$, then $f \leq_c P$.

Because the preceding result really uses c-reductions and not just p-projections, most of our questions on complexity in ideals will be about c-degrees in ideals. Nonetheless, the corresponding questions for p-projections are also interesting.

Bürgisser conjectured that the dependence on the exponent e could be removed:

The Factor Conjecture (Bürgisser [1, Conj. 8.3])**.** *If f is a factor of P, then*

$$C(f) \leq \mathrm{poly}(C(P), \deg f).$$

Although this is how it was phrased in [1], it is natural to extend the Factor Conjecture to conjecture that if f_n is a factor of P_n for all n, then $f \leq_c P$. When we refer to the Factor Conjecture, we may mean either in its original formulation, or in this natural strengthening.

While the Factor Conjecture remains open, Bürgisser proved the dependence on the exponent e in Theorem 3.1 could be removed at the cost of using approximative circuit complexity rather than plain circuit complexity.

Definition 3.2. We say that a polynomial f has *approximative complexity* $\leq s$, or is *infinitesimally approximated by circuits of size* s if there is a family of

circuits C_ε, depending on a formal parameter ε, such that each of these circuits has size at most s, and if we let f_ε be the polynomial computed by C_ε, then $\lim_{\varepsilon \to 0} f_\varepsilon = f$. The *approximative circuit complexity of* f, denoted $\overline{C}(f)$, is the minimum such s.

Theorem 3.3 (Bürgisser [2])**.** *Over any field of characteristic zero, if f is a factor of P, then*

$$\overline{C}(f) \leq \mathrm{poly}(C(P), \deg f).$$

As an additional interesting consequence, we get a polynomial equivalence between the complexity of (approximatively) computing a polynomial f, and deciding whether a point (x, y) is in the graph of f, that is, given x and y, deciding whether $f(x) = y$. The *decision complexity* of a set $S \subseteq \mathbb{F}^n$, denoted $D(S)$, is the minimum number of arithmetic operations and comparison tests ($=, \neq$) sufficient to decide, for a given input $\vec{x} \in \mathbb{F}^n$, whether or not $\vec{x} \in S$. First, note that we trivially have $D(\mathrm{graph}(f)) \leq C(f) + 1$: given x, y, evaluate $f(x)$, then in one more operation check whether $f(x) = y$. In the converse direction, the previous result yields:

Corollary 3.4 (Bürgisser [2])**.** *Over any field of characteristic zero, $\overline{C}(f) \leq \mathrm{poly}(D(\mathrm{graph}(f), \deg f)$. The same holds if we replace deterministic decision tree complexity D with randomized decision complexity two-sided error.*

This result, in addition to its fundamental importance, will play a (small) role in our discussion at the end.

More recently, these factorization theorems were strengthened to several natural algebraic complexity classes (see Appendix A for definitions):

Theorem 3.5. *If f is a factor of P, then:*

1. *Quasi-polynomial closure of ABPs, VP_{ws} [11]:*

$$\mathrm{ABP}(f) \leq \mathrm{poly}(\mathrm{ABP}(P), \deg P, (\deg f)^{O(\log \deg f)})$$

2. *Quasi-polynomial closure of formulas, $\mathsf{VP}_e = \mathsf{VF}$ [11]:*

$$F(f) \leq \mathrm{poly}(F(P), \deg P, (\deg f)^{O(\log \deg f)})$$

3. *Closure of formulas, $\mathsf{VP}_e = \mathsf{VF}$, for polynomials of bounded individual degree [31]:*

$$F_{\Delta+5}(f) \leq \mathrm{poly}(F_\Delta(P), (\mathrm{ideg}(P)n)^{\mathrm{ideg}(P)})$$

4. *Square-root exponential closure of bounded-depth circuits* VAC^0 *[8]:*

$$C_{\Delta+O(1)}(f) \leq \mathrm{poly}(C_\Delta(P), \deg P, (\deg f)^{\sqrt{\deg f}})$$

5. *Closure of* VNP *[8]: if* $P(\vec{x}) = \sum_{\vec{y}\in\{0,1\}^m} Q(\vec{x}, \vec{y})$*, then there exists* g *such that* $f(\vec{x}) = \sum_{\vec{y}\in\{0,1\}^{m'}} g(\vec{x}, \vec{y})$ *where*

$$C(g) \leq \mathrm{poly}(C(Q), \deg P, \deg f).$$

While the following result is not explicitly about factorization nor complexity of ideals, it is the key technical result in [8], and may have further uses in studying complexity in ideals of polynomials.

Theorem 3.6 (Chou, Kumar, and Solomon [8])**.** *Over any field of characteristic zero, let* $P(x_0, \ldots, x_n)$ *and* $f(x_1, \ldots, x_n)$ *be polynomials such that* $P(f(\vec{x}), \vec{x}) = 0$*. Then*

$$C_{\Delta+3}(f) \leq \mathrm{poly}(C_\Delta(P), \deg P, (\deg f)^{\sqrt{\deg f}}).$$

There is also a significant body of literature around algorithms for factorization of multivariate polynomials, some of which has fed into some of the above results. We refer the reader to Kaltofen's survey [23], as well as to some of the more recent results on the equivalence between factoring polynomials and polynomial identity testing [25, 12, 14].

4 Ideals in algebraic circuit lower bounds

Essentially all lower bounds in algebraic complexity theory either use the substitution/restriction method, or a "rank-like" method, or sometimes a combination of the two. See, e. g., the surveys [39, 10]. Rank-like methods take the following form: to each polynomial f we associate a matrix $M(f)$ (which might be exponentially large). For example, a matrix of partial derivatives of f. Then, to show a lower bound like $\mathcal{C}_{hard} \not\subseteq \mathcal{C}_{easy}$, the method proceeds by:

1. Showing that for every $f \in \mathcal{C}_{easy}$, $M(f)$ has low rank
2. Finding some $f_{hard} \in \mathcal{C}_{hard}$ for which $M(f_{hard})$ has high rank.

Thus, the rank of $M(f)$ is used to show that there is some $f_{hard} \in \mathcal{C}_{hard}$ that is not in $\mathcal{C}_{easy}$.

Rank-like methods are particular instances of a more general "polynomial method," as follows. In all examples we are aware of, the entries of

$M(f)$ are in fact polynomials in the coefficients of f (in fact, even just linear combinations thereof). Recall that $\mathrm{rk}\, M(f) < r$ if and only if the $r \times r$ minors (=determinants of $r \times r$ submatrices) vanish. If each entry of $M(f)$ is a polynomial in the coefficients of f, then these minors are also polynomials in the coefficients of f. We refer to such polynomials—whose variables are the coefficients of the polynomials f we are studying—as "meta-polynomials." Let $\mathrm{coeff}(f)$ denote the vector of coefficients of f. The rank method then proceeds by showing that certain meta-polynomials vanish on $\mathrm{coeff}(f)$ for every $f \in \mathcal{C}_{easy}$, and that at least one of these meta-polynomials does not vanish on $\mathrm{coeff}(f_{hard})$.

The more general polynomial method is to then search for such meta-polynomials that can separate the complexity classes, not necessarily restricted to those coming from the rank of $M(f)$ for some M. As mentioned, all lower bounds we are aware of either use the polynomial method or substitution, or both. It is thus reasonable to study the power of the polynomial method, and learn what we can about its structure and limitations. For us, the key observation (going back a century or more) is that the collection of polynomials that vanish everywhere on a given set S (e. g., $S = \mathcal{C}_{easy}$) is an ideal! For if f, g vanish on S, then so does $f+g$, and so does fh for any polynomial h. Studying the complexity of these meta-polynomials thus brings us back to complexity in ideals.

4.1 Algebraic natural proofs

One way of proving lower bounds on certain ideals of (meta-)polynomials is by exhibiting succinct hitting sets; this is the main content of the algebraic natural proofs connection [16, 17], which we discuss next.

We can formalize the above polynomial method in the following definition:

Definition 4.1 ($\mathcal{D}$-natural against $\mathcal{C}$, [17, Definition 4] and [16, Definition 1.1])**.** Let $\mathcal{C}$ be a class of polynomials, and $\mathcal{D}$ a class of meta-polynomials (whose variables are the coefficients of polynomials in $\mathcal{C}$). A property $\Pi = (\Pi_n)$, where Π_n is a collection of polynomials for each n, is *$\mathcal{D}$-natural against $\mathcal{C}$* if it contains a subset $\Pi_n^* \subseteq \Pi_n$ satisfying:

1. *Largeness:* Π_n^* is the complement of the zero-set of a meta-polynomial T_n;
2. *$\mathcal{D}$-Constructivity:* The meta-polynomial family $T = (T_n)$ is in $\mathcal{D}$; and
3. *Usefulness against $\mathcal{C}$:* Any family of functions $f = (f_n)$ with $f_n \in \Pi_n^*$ for all n is not contained in $\mathcal{C}$.

As mentioned above, essentially all known algebraic circuit lower bounds to date are natural in this sense, see, e. g., [19]; Appendix B of *ibid.* also gives a discussion of why this polynomial method may be one of the easiest ways to prove such lower bounds.

Note that Largeness and Usefulness together imply that T_n vanishes on $\{\mathrm{coeff}(f) : f \in \mathcal{C}\}$. Now, since the set of meta-polynomials that vanish on $\mathcal{C}$ is an ideal, the question of whether there is a $\mathcal{D}$-natural property against $\mathcal{C}$ is the same as asking whether the ideal of polynomials that vanish on $\mathcal{C}$ contains any families in $\mathcal{D}$.

We now come to the connection with succinct hitting sets.

Definition 4.2 (Succinct hitting set [17, Definition 5] and [16, Definition 1.3])**.** A set $\mathcal{C}$ of families of polynomials is a *succinct hitting set* against a set of meta-polynomials $\mathcal{D}$ if for every nonzero $T \in \mathcal{D}$, there is some $f \in \mathcal{C}$ such that $T(\mathrm{coeff}(f)) \neq 0$.

It is then straightforward to observe:

Theorem 4.3 (Algebraic natural proofs [17, Theorem 1] and [16, Theorem 1.4])**.** *For any two algebraic complexity classes $\mathcal{C}, \mathcal{D}$, there is no property which is $\mathcal{D}$-natural against $\mathcal{C}$ if and only if $\mathcal{C}$ is a succinct hitting set against $\mathcal{D}$.*

A central question thus arises:

Open Question 4.4 ([16, 17])**.** *Is* VP *a succinct hitting set against* VP*? Conversely, do there exist meta-polynomials that vanish on* VP *and that can be computed by polynomial-sized circuits?*

While this question still remains open, Forbes, Shpilka, and Volk [16] upgraded many of the known hitting sets in the literature to succinct hitting sets, thereby proving lower bounds on the corresponding ideals. Here we briefly summarize many of their results; for notation not defined in Appendix A, we refer to their paper [16].

Theorem 4.5 (Forbes–Shpilka–Volk [16])**.** *Let I_n be the ideal whose vanishing defines the set of $\mathrm{poly}(\log s, n)$-size multilinear $\Sigma\Pi\Sigma$ formulas, within the space of all multilinear formulas on n variables. Then every family $f = (f_n) \in I = (I_n)$ is not computable by size-s computations of the form*

- $\Sigma^{O(1)}\Pi\Sigma$ *formulas*
- $\Sigma\Pi\Sigma$ *formulas of transcendence degree $O(1)$*
- *Sparse polynomials (=$\Sigma\Pi$ circuits)*

- $\Sigma m \wedge \Sigma\Pi^{O(1)}$ *formulas*
- *Commutative roABPs*
- *Constant-depth, constant-occur formulas*
- *Arbitrary circuits composed with sparse polynomials of transcendence degree* $O(1)$

Remark 4.6. Geometric Complexity Theory (GCT) (see, e. g., [29, 28] and references therein and thereto) also aims to produce such meta-polynomials, but to find them by additionally taking advantage of the symmetries most complexity classes have. For example, most complexity measures do not change under permutation of the variables, leading to an action of the symmetric group S_n on the corresponding complexity classes. Many complexity measures do not change significantly under invertible linear combinations of the variables, leading to an action of the general linear group $\mathrm{GL}_n(\mathbb{F})$. These actions organize the meta-polynomials into *representations* of the corresponding groups, and we may now ask about the circuit complexity of certain representation-theoretically-defined polynomial families. See [17, Section 4.1] for details.

5 Algebraic proof complexity

While proof complexity is now a vast field of its own, the question which perhaps defined the field is: given an unsatisfiable Boolean formula φ, what is the length of the shortest proof of $\neg\varphi$ in a standard, line-by-line (Frege) deduction system? This very natural question, seemingly limited in scope, leads to a field with connections to many others. From the point of view of Logic, one can see proof complexity as being about the strength of different proof systems for proving tautologies (equivalently, refuting unsatisfiable formulas). From the point of view of Structural Complexity, proof complexity lower bounds are stepping stones towards proving $\mathsf{NP} \neq \mathsf{coNP}$. From the point of view of Algorithms, proof complexity lower bounds can prove limitations on some of our current best practical SAT solvers. We encourage the reader to consult some of the excellent surveys such as [5, 34, 30, 36] for more, but for now we want to walk briskly from here to complexity in ideals of polynomials.

One approach to proof complexity which seems to differ from the standard, line-by-line deduction systems is the following algebraic approach. (Though, in a surprising twist, Li, Tzameret, and Wang [26] proved that

a noncommutative version of this algebraic approach is actually quasi-polynomially *equivalent* to standard Frege deduction systems!) First, we translate Boolean formulas φ into "equivalent" systems of polynomial equations, such that the roots of the system of equations are in bijective correspondence with the satisfying assignments to φ. This is achieved by the following standard transformation:

Boolean	algebraic
x	$1 - x$
$\neg f$	$1 - T(f)$
$f \vee g$	$T(f) \cdot T(g)$

Note that the Boolean formula consisting of a single variable x is satisfied if and only if $x = 1$ if and only if the polynomial equation $1 - x = 0$ is satisfied. Similarly, $T(f)T(g) = 0$ if and only if $T(f) = 0$ or $T(g) = 0$, thus multiplication of polynomials corresponds to disjunction of Boolean formulas. To force the only solutions to our system of equations to be $\{0, 1\}$-valued, we also include the equations $x_i(1 - x_i) = 0$ for each variable x_i.

Now, by the correspondence above, we have that φ is unsatisfiable if and only if the system of equations just described, which we'll denote F_φ, has no roots. (Because any roots must be $\{0, 1\}$-valued by construction, we can work over any field $\mathbb{F}$ we like, though complexity results may depend on the field.) We are then aided by the following germinal theorem (see, e. g., Eisenbud [13] for a textbook treatment):

Theorem 5.1 (Hilbert's Nullstellensatz [20])**.** *Let $\mathbb{F}$ be an algebraically closed field. Then a system of equations $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ has no solution if and only if 1 is in the ideal $\langle f_1, \ldots, f_m \rangle \subseteq \mathbb{F}[\vec{x}]$.*

There are several different proof systems based on this principle [6, 4, 32, 33, 18], with the current strongest being the Ideal Proof System (IPS), first introduced by Pitassi [32, 33] and expanded upon by Grochow & Pitassi [18]. IPS starts by introducing new placeholder variables y_i for the equations we are showing unsatisfiable.

Definition 5.2 (Ideal Proof System [32, 33, 18])**.** An *IPS* certificate that a system of polynomial equations over a field $\mathbb{F}$, $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ is unsatisfiable over the algebraic closure $\overline{\mathbb{F}}$ is an $\mathbb{F}$-polynomial $C(x_1, \ldots, x_n, y_1, \ldots, y_m)$ such that

1. $C(\vec{x}, \vec{0}) = 0$, and

2. $C(\vec{x}, \vec{f}(\vec{x})) = 1$.

The first condition here is equivalent to C being in the ideal generated by the y_i. Combining with the second condition, this then implies that 1 is in the ideal generated by the f_i, and thus that the system of equations is unsatisfiable, by the Nullstellensatz. Although there is no "line-by-line" deduction here (intermediate gates in C may computed polynomials outside the ideal of the y_i), an IPS certificate nonetheless serves as a proof that the system of equations is unsatisfiable.

The fact that C is just an ordinary circuit, and *not* a line-by-line deduction, is partly what creates a close relationship between IPS and ordinary algebraic circuit lower bounds. Namely, super-polynomial lower bounds on IPS imply that $\mathsf{VP} \neq \mathsf{VNP}$ [18]. We expect such lower bounds to hold, because if $\mathsf{NP} \not\subseteq \mathsf{coMA}$, then some tautologies (after being translated into polynomial systems as above) require super-polynomial-sized IPS proofs [32].

To draw the connection with complexity in cosets of ideals, we rephrase the definition of IPS certificate in terms of ideals. We've already mentioned that condition (1) of the definition implies that $C \in \langle y_1, \ldots, y_m \rangle$. Condition (2) of the definition says that C is congruent to 1 modulo the ideal $\langle y_1 - f_1(\vec{x}), \ldots, y_m - f_m(\vec{x}) \rangle$, as modding out by this ideal is the same as substituting in the $F_i(\vec{x})$ in place of the y_i.

Thus, the set of IPS certificates is a coset of an ideal:

$$(1 + \langle y_1 - f_1(\vec{x}), \ldots, y_m - f_m(\vec{x}) \rangle) \cap \langle y_1, \ldots, y_m \rangle.$$

(The intersection of two ideal cosets is either empty or a coset of an ideal.) The question of lower bounds on IPS is thus precisely a question of the complexity of polynomial families in this ideal coset. We will refer to this ideal coset as the "coset of IPS certificates."

As in the case of algebraic natural proofs, Forbes, Shpilka, Tzameret, and Wigderson showed lower bounds on restricted forms of IPS for certain simple families of unsatisfiable systems of equations:

Theorem 5.3 (Forbes–Shpilka–Tzameret–Wigderson [15])**.** *Any $\Sigma\wedge\Sigma$ circuit computing an IPS certificate for the following systems of unsatisfiable equations must have at least exponential size:*

1. $\sum x_i y_i - \beta, \{x_i^2 - x_i\}, \{y_i^2 - y_i\}$*, where* $\beta > n$.
2. $x_1 x_2 \cdots x_n - 1, \sum x_i - m, \{x_i^2 - x_i\}$

Similarly, any read-once algebraic branching program computing an IPS certificate for the following systems of equations must have exponential size:

1. $\sum z_{ij}x_ix_j - \beta, \{x_i^2 - x_i\}, \{z_{ij}^2 - z_{ij}\}$, *where* $\beta > \binom{2n}{2}$.

2. $1 + \prod \left(z_{ij}(x_i + x_j - x_ix_j) + (1 - z_{ij})\right), \{x_i^2 - x_i\}, \{z_{ij}^2 - z_{ij}\}$.

This furnishes our last known examples of lower bounds on polynomials in a coset of an ideal. It is worth noting that all these examples are essentially one equation together with the Boolean axioms. It would be interesting to extend these results to more complicated systems of equations.

6 Open questions

As a starting point for better understanding complexity of polynomials in (coset of) ideals, I'd like to propose several open questions. Of course, most of these fall back in one way or another on Bürgisser's Factor Conjecture. While the Factor Conjecture itself remains an important question, I would still be very happy to see any of these questions resolved either assuming the Factor Conjecture, or with some of the same caveats that appear in previous results (such as those in Section 3).

It is worth emphasizing the following:

> **Every one of these questions is even open for ideals with only 2 generators.**

How far is 2 generators from the general case? Naively, one might expect ideals in n variables to be generated by at most n polynomials, since—if we consider the zeros of the polynomials—each additional polynomial "should" reduce the dimension of the set of solutions by 1. However, this is in fact quite far from the truth. There are examples [37, 38] of ideals in n variables, generated by polynomials of degree $\leq n$, such that the minimum number of generators needed is a function in $\mathcal{E}^{n+1}$, the $(n+1)$-st level of the Grzegorczyk hierarchy, and $n+1$ is sharp. To recall, $\mathcal{E}^1$ is all linear functions, $\mathcal{E}^2$ is all polynomials, $\mathcal{E}^3$ contains all towers of exponentials of fixed height such as $2^{2^{2^n}}$, and in general $\mathcal{E}^{n+1}$ is gotten from $\mathcal{E}^n$ by adding on more layer of primitive recursion. For the closures of complexity classes (Zariski or Euclidean closure, as in Section 4) or for cosets of IPS certificates (Section 5) probably the number of generators needed is much closer to this upper bound than it is to 2, yet even for 2-generated ideals we know almost nothing. My hope is that we will learn structural results about ideals in general that might be useful for understanding these more complicated examples.

In standard circuit complexity, we are aided by the fact that many natural complexity classes have complete problems. Thus, for example, to prove $\mathsf{VP} \neq \mathsf{VNP}$, it suffices to show that the permanent is not in VP. If an ideal I (or coset) had a unique "easiest" polynomial family f (say, of minimum c-degree), then to prove that every polynomial family in I was not in some class $\mathcal{C}$, it would suffice to show that $f \notin \mathcal{C}$, thus returning us to the "ordinary", seemingly easier world of proving circuit lower bounds on an individual polynomial family.

I do not actually have significant hope that this will hold in general. But when it does it should be useful, and answering questions about it will hopefully help us learn more about the structure of c-degrees in ideals, in a way that the techniques involved might help in other ways.

What can we say about the possibility of minimum c-degrees? Well, of course, assuming the Factor Conjecture, every principal ideal family $I_n = \langle f_n \rangle$ has $f = (f_n)$ as its unique minimum c-degree. It is also not hard to construct "uninteresting" non-principal ideals with unique minimum c-degree:

Observation 6.1. *Let $f = (f_n)$ be a multivariate polynomial family such that the factorization of f_n into irreducibles is square-free. Let $X^{(1)}, X^{(2)}, X^{(3)}, \ldots$ denote disjoint sets of variables. Then, over a sufficiently large field, the ideal family $I_n = \langle f_n(X^{(1)}), f_n(X^{(2)}), \ldots \rangle$ has f as its unique minimum c-degree, assuming the Factor Conjecture.*

Proof. Let $g_n \in I_n$ for all n; We will give a c-reduction from $f = (f_n)$ to $g = (g_n)$. Any polynomial $g_n \in I_n$ is of the form $\sum_i f_n(X^{(i)}) g_n^{(i)}(X^{(1)}, X^{(2)}, \ldots)$. Without loss of generality, let us assume that we have written this so that $g_n^{(1)}$ is not in the ideal generated by $f_n(X^{(2)}), f_n(X^{(3)}), \ldots$. Let C be a point such that $f_n(C) = 0$ and $g_n^{(1)}(X^{(1)}, C, C, \cdots, C)$ is not identically zero; such a C exists over any sufficiently large field, since $g_n^{(1)} \notin \langle f_n(X^{(2)}), f_n(X^{(3)}), \ldots \rangle$ by construction, and the latter ideal is radical, by the assumption of square-freeness of f (this guarantees that if $g_n^{(1)}$ is not in the ideal, then neither is any power of it). Then setting $X^{(2)} = X^{(3)} = \cdots = C$, we get that $g(X^{(1)}, C, C, \ldots, C) = f_n(X^{(1)}) g_1(X^{(1)}, C, C, \ldots, C)$, of which $f_n(X^{(1)})$ is a factor. □

To avoid the above kind of triviality, we add an intentionally vague adjective to the following question:

Open Question 6.2. *Are there interesting non-principal ideals with a unique minimum c-degree?*

In particular, we conjecture the following positive example:

Conjecture 6.3. *For each n, let X be an $n \times n$ matrix with independent variable entries x_{ij}, and let I_n be the ideal generated by all $\binom{n}{n/2}^2$ minors of X of size $(n/2)\times(n/2)$. Then the ideal family (I_n) has the determinant as its unique minimum c-degree.*

The main reason we think this might hold is the close connection between matrix rank and determinants; in particular, a matrix X has rank $< r$ if and only if the determinant of every $r \times r$ submatrix vanishes. In general, computing the rank of a matrix should have the same complexity as computing the determinant; at the very least, we can say that testing whether $\det(X) = 0$ (equivalently, testing whether X has full rank) is essentially equivalent to computing $\det(X)$, by Corollary 3.4.

One might also conjecture the analogous results for the permanent and the ideal of all permanents of $n/2\times n/2$ sub-matrices. While this is certainly an interesting question worth answering, I feel less certain about which way it should resolve, precisely because we have no such connection as we have in the determinant case. (Though this is still the $n/2$-nd Jacobian ideal of the permanent hypersurface.)

Conversely, we also ask:

Open Question 6.4. *Must* every *ideal family have a unique minimum c-degree?*

Surely the answer here is negative, but we have not yet constructed a counterexample.

However, even if there is not a *unique* minimum c-degree, from the perspective of proving lower bounds we would still be in good shape if there were only finitely many. Here we recall that a polynomial family f is of *minimal* (not minim*um*) c-degree in an ideal family I if any $f' \in I$ such that $f' \leq_c f$ in fact has $f' \equiv_c f$.

Open Question 6.5. *Does every ideal have only finitely many minimal c-degrees?*

Open Question 6.6. *Is it the case that every ideal has at least one minimal c-degree? Can an ideal have an infinite descending chain of c-degrees?*

Next, instead of considering minimal c-degrees in an ideal, we consider what happens near the "top" of the c-degree structure of an ideal. First, we note that there is no "top," as a family of ideals can contain arbitrarily complicated families:

Proposition 6.7. *Assume the (circuit reduction) Factor Conjecture. Let $c \in \mathsf{VP}$, and let I be any ideal in unboundedly many variables. Then the ideal coset $c + I$ is dense among all c-degrees realized by p-bounded families. That is, for every such ideal coset $c + I$ in unboundedly many variables, and every c-degree d of a p-bounded family (not necessarily occurring in $c+I$), there is some c-degree $d' \geq d$ such that d' occurs in $c + I$.*

We note that the ideal cosets arising as cosets of IPS certificates have $c_n = 1$ for all n, so they satisfy the hypothesis of this result.

Proof. We assume the Factor Conjecture. First we show the result for ideals. Let I be a family of ideals, and let g be a p-bounded family of polynomials, not necessarily in I; we'll show that the some c-degree at least that of g occurs in I. We assume for simplicity that g_n and I_n are both in $R_n = \mathbb{F}[x_1, \ldots, x_{\mathrm{poly}(n)}]$. (If this is not the case it can easily be remedied because we assume g is p-bounded.)

Now, let f be a family in I, and define $h = (h_n)$ as $h_n = f_n g_n$. Since $f_n \in I_n$, we have that $h_n \in I_n$ as well. And since g_n is a factor of h_n, by the Factor Conjecture we get that $g \leq_c h$. Thus a c-degree at least that of g occurs in I.

Now, suppose $c + I$ is an ideal coset as in the statement of the proposition, and g is any p-bounded family of polynomials. Let $f \in I$, define h as before, and define $\overline{h}_n = c_n + h_n$. Since c is in VP, we have that $h \leq_c \overline{h}$, since our circuit reduction can simply compute c_n in polynomial size and then subtract it from $\overline{h}_n$ to get h_n. Since we have $g \leq_c h$ (as before), by transitivity we get $g \leq_c \overline{h}$, as desired. □

The structure of the preceding proof suggests that once we go up high enough in the poset of c-degrees in an ideal I, everything is possible. This suggests to explore the bottom of the top, or the last places (as we go up the poset of c-degrees) where it's not the case that absolutely anything can happen. Towards this end, we introduce the following definition:

Definition 6.8 ([18, Question 7.4])**.** Let $c + I$ be a family of cosets of ideals of polynomials. We say that a c-degree d is *satured* in $c + I$ if every $d' \geq_c d$ occurs in $c + I$.

As we imagine ourselves climbing up the poset of c-degrees in an ideal, once we've hit a saturated degree we can stop. In ideals, this really lets us focus on the lowest c-degrees:

Proposition 6.9. *Every c-degree realized by a p-bounded family in an ideal is saturated, assuming (the circuit reduction version of) the Factor Conjecture.*

Proof. We assume the Factor Conjecture. Let I be a family of ideals, $f \in I$ a family of polynomials. Let g be a family of polynomials, not necessarily in I, such that $f \leq_c g$; we'll show that the degree of g occurs in I. More specifically, let $r(n)$ be a polynomial such that $C^{g_{r(n)}}(f_n) \leq \mathrm{poly}(n)$. Let $h = (h_n)$ be the family defined by $h_n = f_n g_{r(n)}$. Since both $f \leq_c g$ and $g \leq_c g$, we get that $h \leq_c g$. Conversely, since $g_{r(n)}$ is a factor of h_n, by the Factor Conjecture we get that $g \leq_c h$. Thus $g \equiv_c h$, and $h \in I$, so the degree of g occurs in I. □

However, in ideal cosets we no longer know this to be the case. We thus ask:

Open Question 6.10 ([18, Question 7.4])**.** *In every ideal coset, is it the case that every c-degree is saturated? Or at least that every c-degree is below some saturated c-degree in the coset?*

Finally, we note that most complexity classes are usually the image of a pretty simple polynomial map, an observation that goes back to Raz [35] (see also [19, Section 3 & Appendix B.2] for further discussion), leading to Raz's definition and proposal to use so-called elusive functions to prove lower bounds. Can we use this observation to get some mileage when thinking about the ideals that vanish on these complexity classes?

We hope we have convinced the reader of the interest and utility of studying complexity in ideals of polynomials (and their cosets), and that some of our readers will try their hand at some of the many open questions proposed here.

A Other algebraic complexity measures

We use $C_d(f)$ to denote the size of the smallest circuit of depth $\leq d$ computing f. A subscript on other complexity measures restricts depth similarly.

A $\Sigma\Pi\Sigma$ circuit is a depth-3 circuit with a linear combination gate at the output (which may have fan-in more than 2, and may have nontrivial, but still constant, coefficients in the linear combination), preceded by a layer of product gates (of unbounded fan-in), preceded by a layer of linear combinations of the inputs. Superscripts in such notation are used to restrict the fan-in, e. g. $\Sigma^{\leq d}\Pi\Sigma$ means a linear combination gate of fan-in at most d at the output. A $\wedge$ may be used in place of Π, in which case instead of a general product gate, only powering gates are used. For example, $\Sigma^{O(1)}\wedge^d\Sigma$ means a constant fan-in linear combination at the output, preceded by powering gates (which take their inputs to the d-th power), preceded by linear combinations of the input variables. A Σm denotes a linear combination gate

where the coefficients in the linear combination are allowed to be monomials.

A *formula* is a formula in the usual sense: using $+, \times$ and parentheses. The size of a formula is the number of $+, \times$ it uses. Formulas may equivalently be viewed as algebraic circuits in which the underlying directed acyclic graph is in fact a tree. The *formula size* of f, denoted $F(f)$, is the size of the smallest formula computing f. A family of polynomials (f_n) is in VP_e ("e" for "expression", a synonym for "formula" in this context) or VF if it is p-bounded and $F(f_n) \leq \mathrm{poly}(n)$. As with circuit size, $F_d(f)$ denotes the size of the smallest formula of depth at most d computing f.

An *algebraic branching program* or *ABP* is a directed acyclic graph with a single source node s and a single sink node t, where each edge has a weight which is an affine linear combination $a_0 + \sum_i a_i x_i$ of the input variables x_i (where each $a_i \in \mathbb{F}$, and the weights on different edges can differ). An ABP "computes" the sum over all directed $s \to t$ paths of the product of the edge weights on that path. The size of an ABP is its number of nodes. We write $ABP(f)$ for the size of the smallest ABP computing f. It is well-known that $C(f) \leq ABP(f)^{O(1)} \leq F(f)^{O(1)}$, and conversely, $F(f) \leq ABP(f)^{O(\log ABP(f))}$ and $ABP(f) \leq C(f)^{\log C(f)}$ (the latter follows from the fact that $\mathsf{VP} = \mathsf{VNC}^2$ [41]). Up to polynomials, $ABP(f)$ is equivalent to the determinant complexity of f, the size of the smallest skew circuit computing f, and the size of the smallest weakly-skew circuit computing f (Malod & Portier [27] and references therein).

A *read-once oblivious ABP* is a directed acyclic graph where the internal nodes are partitioned into layers $V_0, \ldots, V_N$ with $V_0 = \{s\}$ (the source), $V_N = \{t\}$ (the sink), and such that every edge goes from V_{i-1} to V_i for some i, and there is a permutation $\pi \in S_N$ such that the edges in layer i are each labeled by a polynomial of degree at most d in the variable $x_{\pi(i)}$. The size of the smallest read-once ABP computing f is denoted $roABP(f)$.

References

[1] Peter Bürgisser. *Completeness and reduction in algebraic complexity theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer-Verlag, Berlin, 2000. doi:10.1007/978-3-662-04179-6.

[2] Peter Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004. doi:10.1007/s10208-002-0059-5.

[3] Peter Bürgisser. Cook's versus Valiant's hypothesis. *Theoret. Comput. Sci.*, 235(1):71–88, 2000. Selected papers in honor of Manuel Blum (Hong Kong, 1998). doi:10.1016/S0304-3975(99)00183-8.

[4] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower Bounds on Hilbert's Nullstellensatz and Propositional Proofs. *Proceedings of the London Mathematical Society*, s3-73(1):1–26, 07 1996. doi:10.1112/plms/s3-73.1.1.

[5] Paul Beame and Toniann Pitassi. Propositional proof complexity: past, present, and future. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific Publishing, 2001.

[6] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. Using the groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 174–183, New York, NY, USA, 1996. Association for Computing Machinery. doi:10.1145/237814.237860.

[7] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Tighter connections between derandomization and circuit lower bounds. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 645–658, Dagstuhl, Germany, 2015. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.APPROX-RANDOM.2015.645.

[8] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Some closure results for polynomial factorization and applications. arXiv:1803.05933 [cs.CC], 2018.

[9] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure of VP under taking factors: a short and simple proof. arXiv:1903.02366 [cs.CC], 2019.

[10] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Found. Trends Theor. Comput. Sci.*, 6(1-2), 2010. doi:10.1561/0400000043.

[11] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: uniform closure results for algebraic classes under factoring. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1152–1165, 2018. arXiv:1710.03214 [cs.CC]. doi:10.1145/3188745.3188760.

[12] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J.*

Comput., 39(4):1279–1293, 2009/10. Originally appeared in STOC '08. doi:10.1137/080735850.

[13] David Eisenbud. *Commutative algebra*, volume 150 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1995. With a view toward algebraic geometry. doi:10.1007/978-1-4612-5350-1.

[14] Michael A. Forbes and Amir Shpilka. Complexity theory column 88: Challenges in polynomial factorization. *SIGACT News*, 46(4):32–49, December 2015. doi:10.1145/2852040.2852051.

[15] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity (CCC '16)*, volume 50 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 32, 17. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2016. doi:10.4230/LIPIcs.CCC.2016.32.

[16] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory Comput.*, 14:Article 18, 45, 2018. Originally appeared in STOC '17. doi:10.4086/toc.2018.v014a018.

[17] Joshua A. Grochow, Mrinal Kumar, Michael Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. ECCC Tech. Report TR17-009 and arXiv:1701.01717 [cs.CC], 2017.

[18] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65:37, 2018. Preliminary version appeared in FOCS 2014 (doi:10.1109/FOCS.2014.20). Also available as arXiv:1404.3820 [cs.CC] and ECCC Technical Report TR14-052. doi:10.1145/3230742.

[19] 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.

[20] David Hilbert. Über die vollen Invariantensysteme. *Math. Ann.*, 42(3):313–373, 1893. doi:10.1007/BF01444162.

[21] Maurice Jansen and Rahul Santhanam. Stronger lower bounds and randomness-hardness trade-offs using associated algebraic complexity classes. In *29th International Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 14 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages 519–530. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012. doi:10.4230/LIPIcs.STACS.2012.519.

[22] Erich Kaltofen. Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC87*, pages 443–452, 1987. doi:10.1145/28395.28443.

[23] Erich Kaltofen. Polynomial factorization: a success story. In *Symbolic and Algebraic Computation, International Symposium ISSAC 2003, Drexel University, Philadelphia, Pennsylvania, USA, August 3-6, 2003, Proceedings*, pages 3–4, 2003. doi:10.1145/860854.860857.

[24] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complexity*, 13(1-2):1–46, 2004. Originally appeared in STOC '03. doi:10.1007/s00037-004-0182-6.

[25] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and polynomial factorization. *Comput. Complexity*, 24(2):295–331, 2015. Special issue from CCC '14. doi:10.1007/s00037-015-0102-y.

[26] Fu Li, Iddo Tzameret, and Zhengyu Wang. Characterizing propositional proofs as noncommutative formulas. *SIAM J. Comput.*, 47(4):1424–1462, 2018. Originally appeared in CCC '15. doi:10.1137/16M1107632.

[27] Guillaume Malod and Natacha Portier. Characterizing Valiant's algebraic complexity classes. *J. Complexity*, 24(1):16–38, 2008. doi:10.1016/j.jco.2006.09.006.

[28] Ketan D. Mulmuley. On P vs. NP and Geometric Complexity Theory: Dedicated to Sri Ramakrishna. *J. ACM*, 58(2), April 2011. doi:10.1145/1944345.1944346.

[29] Ketan D. Mulmuley. The GCT Program toward the P vs. NP problem. *Commun. ACM*, 55(6):98–107, June 2012. doi:10.1145/2184319.2184341.

[30] Jakob Nordström. A (biased) proof complexity survey for SAT practitioners. In Carsten Sinz and Uwe Egly, editors, *Theory and Applications of Satisfiability Testing – SAT 2014*, pages 1–6, Cham, 2014. Springer International Publishing. doi:10.1007/978-3-319-09284-3_1.

[31] Rafael Oliveira. Factors of low individual degree polynomials. *Comput. Complexity*, 25(2):507–561, 2016. doi:10.1007/s00037-016-0130-2.

[32] 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 to 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.

[33] 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. URL: https://www.emis.de/journals/DMJDMV/xvol-icm/14/Pitassi.MAN.html.

[34] Toniann Pitassi and Iddo Tzameret. Algebraic proof complexity: Progress, frontiers and challenges. *ACM SIGLOG News*, 3(3):21?43, August 2016. doi:10.1145/2984450.2984455.

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

[36] Alexander Razborov. Guest column: Proof complexity and beyond. *SIGACT News*, 47(2):66–86, June 2016. doi:10.1145/2951860.2951875.

[37] A. Seidenberg. On the length of a Hilbert ascending chain. *Proc. Amer. Math. Soc.*, 29:443–450, 1971. doi:10.2307/2038577.

[38] Stephen G. Simpson. Ordinal numbers and the Hilbert basis theorem. *J. Symbolic Logic*, 53(3):961–974, 1988. doi:10.2307/2274585.

[39] 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.

[40] L. G. Valiant. Completeness classes in algebra. In *Conference Record of the Eleventh Annual ACM Symposium on Theory of Computing (STOC '79)*, pages 249–261. ACM, New York, 1979. doi:10.1145/800135.804419.

[41] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. doi:10.1137/0212043.