Hitting Sets and Reconstruction for Dense Orbits in $\mathsf{VP_e}$ and $\Sigma\Pi\Sigma$ Circuits

Dori Medini ✉
StarkWare Industries Ltd., Netanya, Israel

Amir Shpilka ✉
Blavatnik School of Computer Science, Tel Aviv University, Israel

Abstract

In this paper we study polynomials in $\mathrm{VP_e}$ (polynomial-sized formulas) and in $\Sigma\Pi\Sigma$ (polynomial-size depth-3 circuits) whose orbits, under the action of the affine group $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ (the action of $(A, \boldsymbol{b}) \in \mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ on a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ is defined as $(A, \boldsymbol{b}) \circ f = f(A^T\boldsymbol{x} + \boldsymbol{b})$), are *dense* in their ambient class. We construct hitting sets and interpolating sets for these orbits as well as give reconstruction algorithms. Specifically, we obtain the following results:

1. For $\mathrm{C}_n\left(\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})\right) \triangleq \mathrm{Trace}\left(\begin{pmatrix} \ell_1(\boldsymbol{x}) & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} \ell_n(\boldsymbol{x}) & 1 \\ 1 & 0 \end{pmatrix}\right)$, where the ℓ_is are linearly independent linear functions, we construct a polynomial-sized interpolating set, and give a polynomial-time reconstruction algorithm. By a result of Bringmann, Ikenmeyer and Zuiddam, the set of all such polynomials is dense in $\mathrm{VP_e}$ [14], thus our construction gives the first polynomial-size interpolating set for a dense subclass of $\mathrm{VP_e}$.
2. For polynomials of the form $\mathrm{ANF}_\Delta\left(\ell_1(\boldsymbol{x}), \ldots, \ell_{4^\Delta}(\boldsymbol{x})\right)$, where $\mathrm{ANF}_\Delta(\boldsymbol{x})$ is the canonical read-once formula in *alternating normal form*, of depth 2Δ, and the ℓ_is are linearly independent linear functions, we provide a quasipolynomial-size interpolating set. We also observe that the reconstruction algorithm of [35] works for *all* polynomials in this class. This class is also dense in $\mathrm{VP_e}$.
3. Similarly, we give a quasipolynomial-sized hitting set for read-once formulas (not necessarily in alternating normal form) composed with a set of linearly independent linear functions. This gives another dense class in $\mathrm{VP_e}$.
4. We give a quasipolynomial-sized hitting set for polynomials of the form $f\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)$, where f is an m-variate s-sparse polynomial. and the ℓ_is are linearly independent linear functions in $n \geq m$ variables. This class is dense in $\Sigma\Pi\Sigma$.
5. For polynomials of the form $\sum_{i=1}^{s}\prod_{j=1}^{d}\ell_{i,j}(\boldsymbol{x})$, where the $\ell_{i,j}$s are linearly independent linear functions, we construct a polynomial-sized interpolating set. We also observe that the reconstruction algorithm of [45] works for *every* polynomial in the class. This class is dense in $\Sigma\Pi\Sigma$.

As $\mathrm{VP} = \mathrm{VNC}^2$, our results for $\mathrm{VP_e}$ translate immediately to VP with a quasipolynomial blow up in parameters. If any of our hitting or interpolating sets could be made *robust* then this would immediately yield a hitting set for the superclass in which the relevant class is dense, and as a consequence also a lower bound for the superclass. Unfortunately, we also prove that the kind of constructions that we have found (which are defined in terms of k-independent polynomial maps) do not necessarily yield robust hitting sets.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic complexity, VP, VNP, algebraic circuits, algebraic formula

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.19

Related Version *Full Version*: https://arxiv.org/pdf/2102.05632

Funding The research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 19; pp. 19:1–19:27

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Proving lower bounds on the size of algebraic circuits (also called arithmetic circuits), is an outstanding open problem in algebraic complexity. In spite of much effort, only a handful of lower bounds are known (a detailed account of most known lower bounds can be found in the excellent survey of Saptharishi [61]). One common theme of most known lower bounds is that they are proved using *algebraic arguments*. That is, a proof of a lower bound for a class of circuits $\mathcal{C}$, usually has the following structure: one comes up with a set of (nonzero) polynomials $F_1, \ldots, F_m$, in $N = \binom{n+d}{d}$ many variables, such that the coefficient vector of every n-variate, degree-d polynomial that can be computed in $\mathcal{C}$, is a common zero of all the F_is (such F_is are called *separating polynomials*). Then, one exhibits a polynomial f whose coefficient vector is not a common zero, thus proving $f \notin \mathcal{C}$. As an example one can immediately see that the well known partial derivative technique, and its successor, shifted partial derivative technique, are algebraic. Grochow [29] demonstrated this for most of the known lower bound proofs. As the set of common zeros of a set of polynomials is closed,[1] this immediately implies that if we prove that $f \notin \mathcal{C}$ using an algebraic argument, then the same argument also implies that $f \notin \overline{\mathcal{C}}$, the closure of $\mathcal{C}$. Recall that, in characteristic zero, the closure of a class $\mathcal{C}$ is the set of all polynomials that are limit points of sequences of polynomials from $\mathcal{C}$, where convergence is coefficient-wise (see Definition 9 for a general definition over arbitrary characteristic). As most known techniques are algebraic, we see that for proving a lower bound for a class $\mathcal{C}$ one actually has to consider the larger, and less structured class, $\overline{\mathcal{C}}$.

Geometric Complexity Theory (GCT for short), which was initiated by Mulmuley and Sohoni [55, 56], approaches the lower bound question from a different angle. GCT also looks for an algebraic lower bound proof, but rather than exhibiting an algebraic argument, it aims to prove the existence of a separating polynomial. Specifically, GCT attempts to prove Valiant's hypothesis, that VP≠VNP, over $\mathbb{C}$, via *representation theory*. Valiant's hypothesis is, more or less, equivalent to showing that the permanent of a symbolic $n \times n$ matrix is not a *projection* of the symbolic $m \times m$ determinant for any $m = m(n)$ polynomial in n.[2] Recall that a projection of a polynomial is a restriction of the polynomial to an affine subspace of its inputs. Observe that a restriction of an n-variate polynomial $f(\boldsymbol{x})$ to a subspace of its inputs, is equivalent to considering the polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$, where A is an $n \times n$ matrix and $\boldsymbol{b} \in \mathbb{C}^n$. As any matrix is a limit point of a sequence of invertible matrices, an algebraic proof that the permanent is not a projection of the $m \times m$ determinant, over $\mathbb{C}$, is equivalent to an algebraic proof showing that the permanent is not in the closure of the set of polynomials $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{B} \in \mathbb{C}^{m^2}\}$, where $\mathrm{GL}_m(\mathbb{C})$ is the group of invertible $m \times m$ matrices (this is true for every field of characteristic $\neq 2$). The set $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{B} \in \mathbb{C}^{m^2}\}$ is called the *orbit* of the determinant under the action of the affine group (we denote the affine group over $\mathbb{C}^m$ with $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$). GCT considers the linear space of polynomials that vanish on every coefficient vector in the orbit of the determinant, and similarly the linear space of polynomials that vanish on every coefficient vector in the orbit of the permanent. There is a natural action of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ on those linear spaces, thus defining two representations of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$. GCT wishes

[1] It is closed in the Zariski topology. Over $\mathbb{R}$ or $\mathbb{C}$ this is the same as being closed in the Euclidean topology.

[2] A super-quasipolynomial lower bound would imply that VP≠VNP whereas a super-polynomial lower bound would imply that permanent does not have polynomial-size algebraic formulas or algebraic branching programs.

to find a separating polynomial by showing that some irreducible representation of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ has strictly larger multiplicity when considering the representation corresponding to the determinant. This approach bypasses the barrier given in [28, 30] as it does not exhibit any efficiently computable separating polynomial but rather just proves the existence of one. However, the representation theory questions arising in this program are quite difficult, even when considering the analog questions for restricted classes. For an introduction to GCT see the lecture notes of Bläser and Ikenmeyer [13].

Another possible approach for proving lower bounds against a class of polynomials $\mathcal{C}$, is via the construction of a *hitting set* for $\mathcal{C}$. Recall that a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is a set of points such that for any nonzero polynomial f, that can be computed by a circuit from $\mathcal{C}$, there is $\boldsymbol{v} \in \mathcal{H}$ such that $f(\boldsymbol{v}) \neq 0$. In [37] Heintz and Schnorr observed that if we have such a hitting set $\mathcal{H}$ then any nonzero polynomial g that vanishes on $\mathcal{H}$ cannot be computed in $\mathcal{C}$. It is also not hard to see that this way of obtaining lower bounds also bypasses the natural proof barrier of [28, 30]. The problem is that in most cases we obtained a hitting set for a class only after proving a lower bound for it.

In [26] Forbes and Shpilka defined the notion of a *robust* hitting set for a circuit class $\mathcal{C}$. Over fields of characteristic zero, a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is c-robust if it also satisfies that for every $f \in \mathcal{C}$ there is $\boldsymbol{v} \in \mathcal{H}$ such that $|f(\boldsymbol{v})| \geq c \cdot \|f\|$, where $\|\cdot\|$ is some fixed norm on $\mathbb{C}[\boldsymbol{x}]$ (see Definition 13 for a definition over arbitrary fields). It is not hard to see that if $\mathcal{H}$ is a robust hitting set for a class $\mathcal{C}$ then it also hits the closure of $\mathcal{C}$.

In this work we focus on depth-3 algebraic circuits, known as $\Sigma\Pi\Sigma$, and on $\mathrm{VP_e}$, the class of algebraic formulas, two classes for which we lack strong lower bounds, and in particular we do not have hitting sets for them. For $\Sigma\Pi\Sigma$ circuits the best lower bound is the near cubic lower bound of Kayal, Saha and Tavenas [46], and for $\mathrm{VP_e}$ the best lower bound is the quadratic lower bound of Kalarkoti [39]. Recall that by the result of Valiant et al. [71], a super-quasipolynomial lower bound against $\mathrm{VP_e}$ implies a super-polynomial lower bound against VP. Similarly, a hitting set for $\mathrm{VP_e}$ implies a hitting set for VP. We also note that by a result of Gupta et al. [33], a strong enough lower bound or a hitting set for $\Sigma\Pi\Sigma$ imply both a lower bound for general circuits and a hitting set for them. This result also implies that a polynomial-time reconstruction algorithm for $\Sigma\Pi\Sigma$ circuits would give rise to a sub-exponential time *reconstruction algorithm* for general circuits. Recall that a reconstruction algorithm for a class $\mathcal{C}$ is an algorithm that, given black-box access to a circuit from $\mathcal{C}$, outputs a circuit in $\mathcal{C}$ that computes the same polynomial.

Instead of viewing robust hitting sets as a way to obtain hitting sets for the closure of circuit classes, we suggest to find subclasses of interesting classes, $\tilde{\mathcal{C}} \subset \mathcal{C}$, such that $\mathcal{C}$ is contained in the closure of $\tilde{\mathcal{C}}$, and aim to construct a robust hitting set for the subclass $\tilde{\mathcal{C}}$. This offers a new approach for constructing hitting sets for known classes and for obtaining lower bounds. Specifically, we consider subclasses of $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$ that are dense in their superclasses. Each of these subclasses is the orbit of some simple polynomial under the group of invertible affine transformations.

For $\mathrm{VP_e}$, we first consider a subclass that was defined by Bringmann, Ikenmeyer and Zuiddam [14]–the orbit of the so called *continuant* polynomial (see Definition 27). We give a polynomial-sized interpolating set[3] for this subclass as well as a polynomial-time

[3] Recall that an interpolating set for a class $\mathcal{C}$ of polynomials in n variables, over a field $\mathbb{F}$, is a set of points $\mathcal{H} \subset \mathbb{F}^n$ such that for every $f \in \mathcal{C}$, the list of values $f(\mathcal{H})$ uniquely determines f. See Definition 15.

deterministic reconstruction algorithm that uses as oracle a *root-finding algorithm.*[4] In particular, this implies a polynomial-time randomized reconstruction algorithm, and, in some cases, a polynomial-time deterministic algorithm.

In addition, we exhibit two other subclasses that are dense in $\mathrm{VP_e}$. The first class is defined as the orbit of read-once formulas (ROF for short, see Definition 5) and the second as the orbit of read-once formulas in *alternating normal form* (ROANF for short, see Definition 7). We obtain hitting sets for both classes and an interpolating set for the second. We also observe that the reconstruction algorithm of [35] works for the polynomials in the orbit of ROANFs. Although the results that we obtain for the subclass defined by the continuant polynomial are stronger, we think that every such dense subclass can shed more light on $\mathrm{VP_e}$ and may eventually be used in order to obtain new lower bounds.

For $\Sigma\Pi\Sigma$ we consider two subclasses. One is based on orbits of *sparse* polynomials (polynomials having polynomially many monomials) and the other on orbits of *diagonal* tensors (see Definition 40). We give a hitting set for the first, an interpolation set for the second, and we also observe that a slight modification of the randomized reconstruction algorithm of [43] applies for the second class.

In particular, our results give the first dense subclasses inside $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ for which a polynomial-size interpolating set is known as well as a polynomial-time reconstruction algorithm. By [71] our result immediately translate to VP, giving a dense subclass of for which a quasipolynomial-sized interpolating set is known as well as a quasipolynomial-time reconstruction algorithm.

If we could transform the interpolating sets that we have found to *robust hitting sets* for the orbits, then this will immediately give hitting sets for the closure of the orbits, i.e. for $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$, which, by [37] gives a lower bound for the class. Thus, our work raises an intriguing problem:

▶ **Problem 1.** *Given an interpolating set for a class $\mathcal{C}$ construct a robust hitting set for $\mathcal{C}$.*

We stress that by our results, solving this problem would lead to hitting sets, and lower bounds, for $\mathrm{VP_e}$ and VP.

Another advantage for having small interpolating sets for dense subclasses is the following: One approach for searching for separating polynomials for a class, is by considering the map from circuits in the class to the coefficient vectors of the polynomials that they compute. That is, once we fix a computation graph, an assignment to the constants appearing in the circuit determines the output polynomial. Each coefficient is a polynomial in those constants, and as there are "few" constants (polynomially many for polynomially sized circuits), and there are exponentially many coefficients, there should be many polynomials vanishing on the closure of the image of this map. If we could get a good understanding of this map then perhaps we could use it to construct a polynomial that vanishes on all such coefficient vectors. This polynomial will vanish on all coefficient vectors of the superclass in which the subclass is dense. A different approach is to find a coefficient vector that is not in the closure of the image of this map (this is the approach of Raz in [57]). Now, assume that $\mathcal{H}$ is an interpolating set for a dense subclass $\tilde{\mathcal{C}} \subset \mathcal{C}$. We know that the map $f \to f\big|_{\mathcal{H}}$ is one-to-one on $\tilde{\mathcal{C}}$. Thus, the list of values $f\big|_{\mathcal{H}}$ can be viewed as an efficient encoding that is given in terms of values of the computed polynomial. This provides a different encoding of a circuit – instead of the constants in it, use the evaluations on $\mathcal{H}$. Thus, by studying the closure of

[4] A root-finding algorithm, over a field $\mathbb{F}$, when given black-box access to a univariate polynomial, outputs a root of that polynomial in $\mathbb{F}$, if such a root exists.

this map (i.e. the closure of the set of points on $\mathbb{F}^{|\mathcal{H}|}$ that can be obtained as evaluation vectors of polynomials in the subclass) we may be able to find a separating polynomial, or, as in Raz's approach, find an evaluation vector that is not obtained by any polynomial in the superclass. It is clear that one can also try this approach even if $\mathcal{H}$ is not an interpolating set, however, as interpolating sets "preserve information" of a dense set, we believe that such sets are better suited for this approach.

To conclude, focusing on dense subclasses and studying their properties could lead to better understanding of their superclasses and perhaps to breakthrough results in algebraic complexity.

To formally state our results we need some definitions that we give next.

1.1 Basic definitions

1.1.1 Notation

For $k \in \mathbb{N}$, we denote $[k] \triangleq \{1, 2, 3, \ldots, k\}$ and $[k]_0 \triangleq \{0, 1, 2, \ldots, k-1\}$. We use boldface lowercase letters to denote tuples of variables or vectors, as in $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{a} = (a_1, \ldots, a_m)$, when the dimension is clear from the context. For any two elements i, j coming from some set S (usually i and j will be numbers), $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise.

The individual degree of a variable x_i in $f(\boldsymbol{x})$ is the degree of f as a polynomial in x_i. A polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ of $\deg(f) \leq 1$ is called a linear function, and if f is homogeneous then it is called a *linear form*. For a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and an integer $k \in \mathbb{N}$ we denote by $f^{[k]}$ the degree-k homogeneous part of $f(\boldsymbol{x})$,i.e. the sum of all monomials of f of degree exactly k. In particular,

$$f(\boldsymbol{x}) = f^{[0]}(\boldsymbol{x}) + f^{[1]}(\boldsymbol{x}) + \ldots + f^{[\deg(f)]}(\boldsymbol{x}) \ .$$

Note that for a linear function f, $f^{[1]}$ is a linear form. We say that a polynomial f is homogeneous of degree k or that f is k-homogeneous if $f = f^{[k]}$. We say a set of linear functions $\{\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})\} \subset \mathbb{F}[\boldsymbol{x}]$ is *linearly independent* if the set $\left\{\ell_i^{[1]}\right\}$ is linearly independent.[5] Given a polynomial $f(\boldsymbol{x})$, a subset of variables $\boldsymbol{y} \subseteq \{x_1, \ldots, x_n\}$ and an assignment to those variables $\boldsymbol{a} \in \mathbb{F}^{|\boldsymbol{y}|}$, we denote by $f\big|_{\boldsymbol{y}=\boldsymbol{a}} \in \mathbb{F}[\boldsymbol{x} \setminus \boldsymbol{y}]$ the polynomial resulting from assigning the values of $\boldsymbol{a}$ to the variables of $\boldsymbol{y}$ in $f(\boldsymbol{x})$. We sometimes abuse notation and write $\boldsymbol{y} \subseteq [n]$ to indicate the indices of the assigned variables instead of the variables themselves.

1.1.2 Circuit classes

▶ **Definition 2.** *An algebraic formula (also called arithmetic formula) over a field $\mathbb{F}$, is a rooted tree whose leaves are labeled with either variable or scalars from $\mathbb{F}$, and whose root and internal nodes (called gates) are labeled with either "$+$" (addition) or "$\times$" (multiplication). An algebraic formula computes a polynomial in the natural way. Each leaf computes the polynomial that labels it, and each gate computes either the sum or product of its children, depending on its label. The output of the formula is the polynomial computed at its root. The size of a formula is the number of wires in it. The depth of a formula is the length of the longest simple leaf-root path in it. The formula size of a polynomial f is defined as the smallest size of a formula that outputs f.*

[5] Note that by our definition, x and $x + 1$ are linearly dependent.

A sequence $m(n)$ of natural numbers is called polynomially bounded if there exists a univariate polynomial q such that $m(n) \leq q(n)$ for all n.

The complexity class $\mathrm{VP_e}$ is defined as the set of all families of polynomials $(f_n)_n$, with $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, whose formula size is polynomially bounded.

▶ **Definition 3.** *An arithmetic circuit Φ is a $\Sigma^{[s]}\Pi^{[d]}$ circuit if it is a layered graph of depth-2, has a top gate labeled $+$ with fan-in $\leq s$ and its second layer is comprised entirely of $\times$ gates with fan-in $\leq d$. In other words, $\Sigma^{[s]}\Pi^{[d]}$ compute polynomials of degree d with at most s monomials.*

▶ **Definition 4.** *An arithmetic circuit Φ in n variables is a $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit if it is a layered graph of depth-3, has a top gate labeled $+$ with fan-in $\leq s$, its second layer is comprised entirely of $\times$ gates with fan-in $\leq d$, and its bottom layer is comprised of linear functions in $x_1, \ldots, x_n$. In other words, $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit compute polynomials of the form*

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} (\alpha_{i,j,0} + \sum_{k=1}^{n} \alpha_{i,j,k} x_k) \,.$$

Given a family of circuits $\mathcal{C}$, we will sometime denote it as $\mathcal{C}(\mathbb{F})$ to stress that we allow coefficients to come from the field $\mathbb{F}$. Observe that the definitions of the classes above do not depend on the field and so we can define them over any field of our choice.

▶ **Definition 5.** *An* arithmetic read-once formula *(ROF for short) Φ over a field $\mathbb{F}$ in the variables $\boldsymbol{x} = (x_1, \ldots, x_n)$ is a binary tree T whose leaves are labeled with input variables and a pairs of field elements $(\alpha, \beta) \in \mathbb{F}^2$, and whose internal nodes are labeled with the arithmetic operations $\{+, \times\}$ and a field element $\alpha \in \mathbb{F}$. Each input variable can label at most one leaf. The computation is performed in the following way: A leaf labeled with the variable x_i and with (α, β), computes the polynomial $\alpha x_i + \beta$. If a node v is labeled with the operation $* \in \{+, \times\}$ and with $\alpha \in \mathbb{F}$, and its children compute the polynomials Φ_{v_1} and Φ_{v_2}, then the polynomial computed at v is $\Phi_v = \Phi_{v_1} * \Phi_{v_2} + \alpha$. A polynomial $f(\boldsymbol{x})$ is called a* read-once polynomial *(ROP for short) if $f(\boldsymbol{x})$ can be computed by a ROF.*

▶ **Observation 6.** *Read-once polynomials are always multilinear polynomials.*

We next define formulas in alternating normal form, as was first defined in [35].

▶ **Definition 7** (Section 3.2 in [35])**.** *We say that an arithmetic formula Φ, over $\mathbb{F}$, is in* alternating normal form *(Φ is called an* ANF *for short) if:*

1. *The underlying tree of Φ is a complete rooted binary tree (the root node is called the output node). In particular,* $\mathrm{size}(\Phi) = 2^{\mathrm{depth}(\Phi)+1} - 1$*, where* $\mathrm{size}(\Phi)$ *is the number of nodes in the tree of Φ and* $\mathrm{depth}(\Phi)$ *is the maximum distance of a leaf node from the output node of Φ.*
2. *The internal nodes consist of alternating layers of $+$ and $\times$ gates. In particular, the label of an internal node at distance d from the closest leaf node is $+$ if d is even and $\times$ otherwise. So if the root node is a $+$ node, its children are all $\times$ nodes, its grandchildren are all $+$ etc.*
3. *The leaves of the tree are labeled with linear functions. That is, each leaf is labeled with $\ell(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i$, where each $a_i \in \mathbb{F}$ is a scalar.*

The product depth Δ *of Φ is the number of layers of product gates. The number of leaves of Φ is therefore always 4^{Δ} if the top gate is $+$, and $\frac{1}{2} \cdot 4^{\Delta}$ if the top gate is $\times$.*

The class $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ mentioned in Section 1.2.2 is defined in terms of the following canonical read-once ANF formula (ROANF for short):

▶ **Definition 8** (Notation from Fact 3.4 of [35]). *We denote the canonical ROANF polynomial, of product depth Δ on 4^Δ variables, as $ANF_\Delta(\boldsymbol{x})$. It is defined recursively as follows:*

$$ANF_0(\boldsymbol{x}) = x_1$$
$$ANF_{\Delta+1}(\boldsymbol{x}) = ANF_\Delta\left(\boldsymbol{x}^{(1)}\right) ANF_\Delta\left(\boldsymbol{x}^{(2)}\right) + ANF_\Delta\left(\boldsymbol{x}^{(3)}\right) ANF_\Delta\left(\boldsymbol{x}^{(4)}\right) ,$$

where $\boldsymbol{x}^{(i)}$ is the 4^Δ-tuple of variables $\{x_{(i-1)\cdot 4^\Delta+1}, \ldots, x_{i\cdot 4^\Delta}\}$.

For example, $\mathrm{ANF}_1(\boldsymbol{x}) = x_1x_2 + x_3x_4$.

Observe that any polynomial in $\mathrm{ANF}_\Delta^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ is an ANF according to Definition 7, but not vice versa.

1.1.3 Approximate complexity

The following definition gives sense to the notion of approximation over arbitrary fields. In what follows we let ε be a new formal variable.[6] For a field $\mathbb{F}$ we denote with $\mathbb{F}[\varepsilon]$ the ring of polynomial expressions in ε over $\mathbb{F}$, and with $\mathbb{F}(\varepsilon)$ the fraction field of $\mathbb{F}[\varepsilon]$, i.e. the field of rational expressions in ε.

▶ **Definition 9.** *Let $\mathcal{C}(\mathbb{F})$ be a circuit class over a field $\mathbb{F}$. The closure of $\mathcal{C}$, denoted $\overline{\mathcal{C}(\mathbb{F})}$, is defined as follows: A family of functions $(f_n)_n$, where $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, is in $\overline{\mathcal{C}(\mathbb{F})}$ if there is a polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$, and a family of functions $(g_{m(n)})_n \in \mathcal{C}(\mathbb{F}(\varepsilon))$, with $g_{m(n)} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$, such that for all $n \in \mathbb{N}$,*

$$g_{m(n)}(x_1, \ldots, x_{m(n)}) = f_n(x_1, \ldots, x_n) + \varepsilon \cdot g_{n,0}(x_1, \ldots, x_{m(n)}) , \tag{1}$$

for some polynomial $g_{n,0} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$. Whenever an equality as in (1) holds we say that

$$g_{m(n)} = f_n + O(\varepsilon) \quad \text{or} \quad f_n = g_{m(n)} + O(\varepsilon) .$$

In that case we think of $g_{m(n)}$ as an "approximation" of f_n, and we say that the family $(g_{m(n)})_n$ approximates the family $(f_n)_n$.

Alder [3] have shown that over $\mathbb{C}$ it holds that $(f_n) \in \overline{\mathcal{C}(\mathbb{C})}$, in the sense of Definition 9, if and only if it is in the closure of $\mathcal{C}(\mathbb{C})$ in the usual sense. That is, if for every n there exists a sequence of polynomials $g_{n,k} \in \mathcal{C}(\mathbb{C})$ such that $\lim_{k\to\infty} g_{n,k} = f_n$, where convergence is taken coefficient wise. This result holds over $\mathbb{R}$ as well, see [52, 17].

Finally, we note that every matrix is approximable (in the sense of Definition 9) by a non-singular matrix (which is equivalent to being a limit of a sequence of non-singular matrices, in characteristic zero).

▶ **Observation 10.** *For every $A \in \mathbb{F}^{n\times n}$ there exists a non-singular matrix $B \in \mathbb{F}(\varepsilon)^{n\times n}$ such that $A = B + O(\varepsilon)$.*

[6] Intuitively, one should think of ε as an infinitesimal quantity.

1.1.4 Hitting and interpolating sets

▶ **Definition 11.** *A set of points $\mathcal{H} \subseteq \mathbb{F}^n$ is called a* hitting set *for a circuit class $\mathcal{C}$ (we also say that $\mathcal{H}$* hits *$\mathcal{C}$) if for every circuit $\Phi \in \mathcal{C}$, computing a non-zero polynomial, there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $\Phi(\boldsymbol{a}) \neq 0$.*

We next give the definition of a robust hitting set, a notion first defined in [26]. Here we extend the definition for arbitrary characteristic. We start by giving the definition of [26], over characteristic zero (and focus on $\mathbb{C}$) and then the more general definition.

▶ **Definition 12** (Following Definition 5.1 of [26])**.** *Let $\|\cdot\|$ be some norm on $\mathbb{C}[\boldsymbol{x}]$. A hitting set $\mathcal{H}$ for a circuit class $\mathcal{C} \subseteq \mathbb{C}[\boldsymbol{x}]$ is called* robust *if there exists some constant $c > 0$ such that, for every $0 \neq f \in \mathcal{C}$,[7] there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $|f(\boldsymbol{a})| \geq c \cdot \|f\|$.*

For arbitrary characteristic we use the same approach as in Definition 9.

▶ **Definition 13.** *Let $\mathbb{F}$ be a field of arbitrary characteristic. A hitting set $\mathcal{H} \subset \mathbb{F}^n$ for a circuit class $\mathcal{C}(\mathbb{F})$ is called* robust *if for every circuit $\Phi \in \mathcal{C}(\mathbb{F}(\varepsilon))$ computing a polynomial $f(\boldsymbol{x}) = h(\boldsymbol{x}) + \varepsilon \cdot g(\boldsymbol{x})$, where $h(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ and $g(\boldsymbol{x}) \in \mathbb{F}[\varepsilon][\boldsymbol{x}]$, there exists some $\boldsymbol{a} \in \mathcal{H}$ such that $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$.*

It is not hard to prove using the result of [3] that for $\mathbb{F} = \mathbb{C}$, Definitions 12 and 13 are equivalent.

▶ **Observation 14.** *If $\mathcal{H}$ is a finite robust hitting set for $\mathcal{C}(\mathbb{F})$, then $\mathcal{H}$ hits $\overline{\mathcal{C}(\mathbb{F})}$ as well.*

Proof. Consider $0 \neq f \in \overline{\mathcal{C}(\mathbb{F})}$. By Definition 9 there is $g \in \mathcal{C}(\mathbb{F}(\varepsilon))$, such that $f = g + O(\varepsilon)$. Clearly $g \neq 0$. Let $\boldsymbol{a} \in \mathcal{H}$ be such that $g(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. It follows that $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. In particular, $f(\boldsymbol{a}) \neq 0$. ◀

We next define the notion of an interpolating set.

▶ **Definition 15.** *Let $\mathcal{C}$ be a class of n-variate polynomials. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is called an* interpolating set *for $\mathcal{C}$ if, for every $f \in \mathcal{C}$, the evaluations of f on $\mathcal{H}$ uniquely determine f.*

▶ **Observation 16.** *If $\mathcal{H}$ is a hitting set for $\mathcal{C}(\mathbb{F}) + \mathcal{C}(\mathbb{F}) \triangleq \{\alpha f + \beta g : f, g \in \mathcal{C}, \alpha, \beta \in \mathbb{F}\}$, then $\mathcal{H}$ is an interpolating set for $\mathcal{C}$.*

A common method for designing hitting and interpolating sets is via hitting set generators.

▶ **Definition 17.** *A polynomial mapping $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ is called a* hitting set generator *(or simply a generator) for a circuit class $\mathcal{C}(\mathbb{F})$ if for any non-zero n-variate polynomial $f \in \mathcal{C}$, the k-variate polynomial $f \circ \mathcal{G}$ is non-zero.*

Similarly, we call $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ an interpolating set generator *for a circuit class $\mathcal{C}(\mathbb{F})$ if for any two different n-variate polynomials $f_1, f_2 \in \mathcal{C}$, the k-variate polynomial $(f_1 - f_2) \circ \mathcal{G}$ is non-zero.*

Generators immediately give rise to hitting sets.

▶ **Observation 18.** *Let $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ be a generator for $\mathcal{C}(\mathbb{F})$ such that the individual degree of each coordinate of $\mathcal{G}$ is at most r. Let $W \subset \mathbb{F}$ be any set of size $|W| = d \cdot r + 1$. Let $\mathcal{H} = \mathcal{G}\left(W^k\right)$. Then $\mathcal{H}$ hits every n-variate polynomial $f \in \mathcal{C}$ of degree at most d.*

Proof. As $\mathcal{G}$ is a generator, the k-variate polynomial $f \circ \mathcal{G}$ is nonzero. As its individual degrees are bounded by $d \cdot r$ it follows that at least one of the values in $(f \circ \mathcal{G})\left(W^k\right) = f\left(\mathcal{H}\right)$ is not zero. ◀

[7] We abuse notation and write $f \in \mathcal{C}$ when f is the output of some circuit from $\mathcal{C}$.

1.1.5 k-independent maps

Our constructions rely on polynomial mappings $\mathcal{G}_k$, parameterized by some integer $k \leq n$, with the property that the image of $f \circ \mathcal{G}_k$ contains all projections of f to k variables. We call such a map a k-independent map.

▶ **Definition 19.** *We call a polynomial mapping* $\mathcal{G}(y_1, \ldots, y_t, z_1) : \mathbb{F}^{t+1} \to \mathbb{F}^n$ *a* 1-independent polynomial map *if for every index* $i \in [n]$ *there exists an assignment* $\boldsymbol{a}_i \in \mathbb{F}^t$ *to* $y_1, \ldots, y_t$ *such that the* i*th coordinate of* $\mathcal{G}(\boldsymbol{a}_i, z_1)$ *is* z_1*, and the rest of the coordinates are* 0*. For* $k > 1$*, a polynomial mapping* $\mathcal{G}(y_1, \ldots, y_{tk}, z_1, \ldots, z_k) : \mathbb{F}^{k(t+1)} \to \mathbb{F}^n$ *is called a* k-independent polynomial map *(or a* k*-independent map) if* $\mathcal{G}$ *is a sum of* k *variable-disjoint* 1*-independent polynomial maps. We denote* k*-independent polynomial maps as* $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ *when* k, t *are implicit. The* $\boldsymbol{y}$ *variables are called* control variables*.*

A k*-independent polynomial map* $\mathcal{G}$ *is called* uniform *if all* n *coordinates of* $\mathcal{G}$ *are homogeneous polynomials of the same degree.*

We discuss k-independent maps in more detail in Section 2.

1.1.6 Subgroups of the linear and affine groups and their actions

Given a matrix $A \in \mathbb{F}^{n \times n}$ and a tuple of variables $\boldsymbol{x} = (x_1, \ldots, x_n)$, we denote

$$A\boldsymbol{x} = \left(\sum_{i=1}^{n} A_{1,i} x_i, \sum_{i=1}^{n} A_{2,i} x_i, \ldots, \sum_{i=1}^{n} A_{n,i} x_i \right) .$$

Let $n \geq m \in \mathbb{N}$. For an m-variate polynomial $f(x_1, \ldots, x_m) \in \mathbb{F}[x_1, \ldots, x_m]$, a matrix $A = (A_{i,j})_{i,j=1}^{n} \in \mathbb{F}^{n \times n}$ and a vector $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}^n$, we define the n-variate polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$ to be

$$f(A\boldsymbol{x} + \boldsymbol{b}) \triangleq f\left(\sum_{i=1}^{n} A_{1,i} x_i + b_1, \sum_{i=1}^{n} A_{2,i} x_i + b_2, \ldots, \sum_{i=1}^{n} A_{m,i} x_i + b_m \right) . \tag{2}$$

Note that we ignored the last $n - m$ coordinates of $A\boldsymbol{x} + \boldsymbol{b}$.

We denote with $\mathrm{GL}_n(\mathbb{F})$ the group of invertible $n \times n$ matrices over $\mathbb{F}$, and with $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ the group of invertible affine transformation, i.e. all the maps $\boldsymbol{x} \to A\boldsymbol{x} + \boldsymbol{b}$, where $A \in \mathrm{GL}_n(\mathbb{F})$ and $\boldsymbol{b} \in \mathbb{F}^n$.

For an m-variate polynomial f over $\mathbb{F}$, and $n \geq m$ we denote with $f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ the orbit of f under the natural action of $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$:[8]

$$f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{ f(A\boldsymbol{x} + \boldsymbol{b}) \mid A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n \} .$$

We similarly define $f^{\mathrm{GL}_n(\mathbb{F})}$. More generally, for a class of m-variate polynomials $\mathcal{C}(\mathbb{F})$, we denote the *orbit* of $\mathcal{C}$ under $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ by

$$\mathcal{C}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{ f(A\boldsymbol{x} + \boldsymbol{b}) \mid f \in \mathcal{C},\ A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n \} .$$

We similarly define $\mathcal{C}^{\mathrm{GL}_n(\mathbb{F})}$. When we want to speak about orbits of families of polynomials from $\mathcal{C}(\mathbb{F})$, with arbitrary number of variables, we use the notation $\mathcal{C}^{\mathrm{GL}(\mathbb{F})}$ or $\mathcal{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

[8] To be precise, the action is $((A, \boldsymbol{b}) \circ f)(\boldsymbol{x}) = f(A^T \boldsymbol{x} + \boldsymbol{b})$. This is required in order to make the action a homomorphism, however, for the groups that we consider it does not change the orbit.

▶ **Observation 20.** *For any m variate polynomial $f(x_1, \ldots, x_m)$ and $n \geq m$:*

- *For any $A \in GL_n(\mathbb{F})$ and $d \in \mathbb{N}$, $f^{[d]}(A\boldsymbol{x})$ is the d-homogeneous part of $f(A\boldsymbol{x})$.*
- *For any $A \in GL_n^{aff}(\mathbb{F})$, $f(\boldsymbol{x})$ is irreducible if and only if $f(A\boldsymbol{x})$ is irreducible.*
- *The set of matrices A for which $f(\boldsymbol{x}) = f(A\boldsymbol{x})$ forms a multiplicative subgroup of $GL_n(\mathbb{F})$ and a similar claim holds for $GL_n^{aff}(\mathbb{F})$.*

We next define some special groups that serve as group of symmetries of some of the models that we consider. We first define the group of symmetries of $\mathrm{ANF}_\Delta(\boldsymbol{x})$. We denote with I_k the $k \times k$ identity matrix.

▶ **Definition 21.** *For $m, \Delta \in \mathbb{N}$ such that $m = 2^\Delta$, the* tree-symmetry group *$TR_m(\mathbb{F})$ denotes the automorphisms of a rooted complete binary tree of depth Δ. It is defined recursively as follows.*

- *For $m = 1$, $TR_1(\mathbb{F})$ consists only of the identity matrix.*
- *For $m > 0$, $TR_m(\mathbb{F})$ is generated by matrices of the form*

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad \textit{and} \quad \begin{pmatrix} 0 & I_{\frac{m}{2}} \\ I_{\frac{m}{2}} & 0 \end{pmatrix}$$

where $A, B \in TR_{\frac{m}{2}}(\mathbb{F})$.

▶ **Definition 22.** *For any $m = 4^\Delta$, the* tree-scale group *$TS_m(\mathbb{F})$ is the group generated by elements of $TR_m(\mathbb{F})$ and matrices of the form*

$$\begin{pmatrix} \alpha I_{\frac{m}{4}} & 0 & 0 & 0 \\ 0 & \alpha^{-1} I_{\frac{m}{4}} & 0 & 0 \\ 0 & 0 & \beta I_{\frac{m}{4}} & 0 \\ 0 & 0 & 0 & \beta^{-1} I_{\frac{m}{4}} \end{pmatrix}$$

where $0 \neq \alpha, \beta \in \mathbb{F}$.

The importance of the group $\mathrm{TS}_m(\mathbb{F})$ stems from the fact that it is the symmetry group of ANF_Δ. To intuitively see why this is the case, notice that in any representation of an ANF one may swap children of any node without changing the output polynomial. We call such symmetries "tree-symmetries" and they are captured by the group $\mathrm{TR}_n(\mathbb{F})$. A second source of ambiguity comes from the fact that we can rescale the formula. Recall that the output polynomial is of the form $f_1 \cdot f_2 + f_3 \cdot f_4$ (Definition 7). Clearly, the output does not change if we replace f_1 by, say, $2f_1$ and f_2 by $f_2/2$. Such rescaling symmetries are captured by the group $\mathrm{TS}_n(\mathbb{F})$. Finally, another source for ambiguity comes from the fact that the quadratic polynomials computed at the bottom two layers of the ANF may have different representations. For example,

$$4xy + 4wz = (x + y + w - z) \cdot (x + y - w + z) + (w + z + x - y) \cdot (w + z - x + y) \ .$$

As there is an infinite number of representations for each quadratic polynomial (over infinite fields), we can expect to characterize the symmetries in term of the quadratics computed at the bottom two layers of the ANF.

▶ **Fact 23** (Special case of Theorem 5.43(iii) of [35])**.** *Let $m, \Delta, n \in \mathbb{N}$ such that $m = 4^{\Delta-1} \leq n/4$. Let $f = ANF_\Delta(\ell_1, \ldots, \ell_{4m}) \in ANF_\Delta^{GL_n^{aff}(\mathbb{F})}$. Let $Q = (q_1, \ldots, q_m)$ be the list of quadratic polynomials that are computed at the bottom two layers of the formula $ANF_\Delta(\ell_1, \ldots, \ell_{4m})$. In particular, $f = ANF_{\Delta-1}(q_1, \ldots, q_m)$. If $Q' = (q'_1, \ldots, q'_m)$ is any other m-tuple of quadratic polynomials for which $f = ANF_{\Delta-1}(q'_1, \ldots, q'_m)$ then Q is $TS_m(\mathbb{F})$-equivalent to Q'.*

Next, we define the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

▶ **Definition 24.** *For any $n \in \mathbb{N}$ the* permutation-scale group*, denoted $PS_n(\mathbb{F})$, is the set of all matrices $A \in GL_n(\mathbb{F})$ which are row-permutations of non-singular diagonal matrices with determinant one.*

For example, $\begin{pmatrix} 0 & -2 & 0 \\ 0 & 0 & -1 \\ 1/2 & 0 & 0 \end{pmatrix} \in \mathrm{PS}_3(\mathbb{C})$.

▶ **Definition 25.** *Let $s, d, n \in \mathbb{N}$ such that $n = s \cdot d$. A matrix $A \in GL_n(\mathbb{F})$ is a member of the* tensor permutation-scale group*, denoted $TPS_{s,d}(\mathbb{F})$, if $A = (P \otimes I_d) \cdot B$, where P is an $s \times s$ permutation matrix and $B = \begin{pmatrix} B_1 & 0 & \dots & 0 \\ 0 & B_2 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & B_d \end{pmatrix}$ is a block diagonal matrix such that each block B_i of B satisfies $B_i \in PS_d(\mathbb{F})$.*

For example, for $s = d = 2$ the matrix $A = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$ is in $\mathrm{TPS}_{2,2}(\mathbb{C})$, as for $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and $B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \end{pmatrix}$, we have $A = (P \otimes I_2) \cdot B$, and clearly each block of B is in $\mathrm{PS}_2(\mathbb{C})$.

Another way of defining the group is as follows: index rows and columns of A with pairs $(i, j) \in [s] \times [d]$. Then, $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$ if and only if there exists a permutation $\pi : [s] \to [s]$, and for all $i \in [s]$ permutations $\theta_i : [d] \to [d]$ and constants $\alpha_{i,j}$ satisfying $\prod_{j=1}^{d} \alpha_{i,j} = 1$, such that $A_{(i,j),(i',j')} = \delta_{\pi(i),i'} \cdot \delta_{\theta_i(j),j'} \cdot \alpha_{i,j}$ for all i, j.

We next prove that $\mathrm{TPS}_{s,d}(\mathbb{F})$ is the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$. In other words, we show that $\mathrm{T}_{s,d}(\boldsymbol{x}) = \mathrm{T}_{s,d}(A\boldsymbol{x})$ if and only if $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$. Intuitively, $\mathrm{T}_{s,d}$ admits no symmetries other than the trivial ones: permutations on the product gates, and internal permutation-scale of each product gate such that the product of the scale coefficients is 1. This is exactly captured by the group $\mathrm{TPS}_{s,d}(\mathbb{F})$, which is therefore contained in the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

▶ **Lemma 26.** *Let $s, d, n \in \mathbb{N}$, such that $d > 2$ and $n = s \cdot d$. If $A \in GL_n(\mathbb{F})$ satisfies $T_{s,d}(\boldsymbol{x}) = T_{s,d}(A\boldsymbol{x})$, then $A \in TPS_{s,d}(\mathbb{F})$.*

1.2 Our results

We first give our results for the class $\mathrm{VP_e}$ and then for the class of depth-3 circuits, for which it may be easier to obtain a robust hitting set, or prove super-polynomial lower bounds.

1.2.1 The continuant polynomial

Bringmann, Ikenmeyer and Zuiddam [14] defined the following polynomial (in Remark 3.14 of their paper), which they called the continuant polynomial:

▶ **Definition 27.** *The continuant polynomial on n variables, $C_n(x_1,\ldots,x_n)$, is defined as the* trace *of the following matrix product:*

$$C_n(x_1,\ldots,x_n) \triangleq \mathit{Trace}\left(\begin{pmatrix} x_1 & 1 \\ 1 & 0 \end{pmatrix}\cdot\begin{pmatrix} x_2 & 1 \\ 1 & 0 \end{pmatrix}\cdot\ldots\cdot\begin{pmatrix} x_n & 1 \\ 1 & 0 \end{pmatrix}\right). \tag{3}$$

We denote with $C^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$ such that $f_n \in \mathbb{F}[x_1,\ldots,x_n]$ and for some $m \le n$, $f_n \in C_m^{GL_n^{aff}(\mathbb{F})}$.

A result of Allender and Wang implies that the polynomial $x_1\cdot y_1+\cdots+x_8\cdot y_8$ is not in $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ [4]. Thus, as a computational class it is very weak. However, Theorem 3.12 of [14] states that for every field $\mathbb{F}$ of characteristic different than 2, it holds that

$$\overline{\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}} = \overline{\mathrm{VP_e}}\,. \tag{4}$$

We give a polynomial-size interpolating set for the class $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ as well as a polynomial-time reconstruction algorithm for it. We first state a simple result that gives a hitting set for the class.

▶ **Theorem 28.** *Let $f(x_1,\ldots,x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$, for $m \le n$, and arbitrary $\mathbb{F}$. Then, for any uniform 1-independent polynomial map $\mathcal{G}$ over $\mathbb{F}$, $f\circ\mathcal{G} \neq 0$.*

As immediate corollary we get a hitting set for the class.

▶ **Corollary 29.** *For every field $\mathbb{F}$, there is an explicit hitting set $\mathcal{H}\subset\mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^6\right)$, that hits every $0 \neq f \in C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \ge n^2$.*

▶ **Theorem 30.** *For every field $\mathbb{F}$, there is an explicit interpolating set $\mathcal{H}\subset\mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^{10}\right)$, for $\bigcup_{m=1}^{n} C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \ge n^2$.*

▶ **Theorem 31.** *There is a deterministic algorithm that given $\mathbb{F}$, an integer n, oracle access to a root-finding algorithm over $\mathbb{F}$, and black-box access to a polynomial $f(x_1,\ldots,x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$ (for any $m \le n$), runs in polynomial-time and outputs linear functions $(\ell_1(x_1,\ldots,x_n),\ldots,\ell_m(x_1,\ldots,x_n))$ such that*

$$f(x_1,\ldots,x_n) = C_m\left(\ell_1(\boldsymbol{x}),\ldots,\ell_m(\boldsymbol{x})\right)\,.$$

If $|\mathbb{F}| < n^3$ then the algorithm will make queries from a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \ge n^3$, and it also requires oracle access to a root-finding algorithm over $\mathbb{K}$.

1.2.2 Orbits of read-once formulas

We denote with $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists a ROF Φ, on $m \le n$ variables, such that $f_n(x_1,\ldots,x_n) \in \Phi^{\mathrm{GL}_n(\mathbb{F})}$. Similarly, we denote with $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}[\mathbb{F}]}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists Δ such that $4^\Delta \le n$ and $f_n(x_1,\ldots,x_n) \in \mathrm{ANF}_\Delta^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$.

We first make the following simple observation.

▶ **Theorem 32.** *For every field $\mathbb{F}$, it holds that*

$$ANF^{GL^{aff}(\mathbb{F})} \subsetneq ROF^{GL(\mathbb{F})} \subsetneq VP_e(\mathbb{F})\,. \tag{5}$$

However, when taking closures we get

$$\overline{ANF^{GL^{aff}(\mathbb{F})}} = \overline{ROF^{GL(\mathbb{F})}} = \overline{VP_e(\mathbb{F})}\,. \tag{6}$$

Our main results for ROFs and ROANFs are a construction of a hitting set for the orbit of ROFs, and an interpolating set for the orbit of ROANFs. Both constructions are obtained using independent polynomial maps (Definition 19).

▶ **Theorem 33.** *Let $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$ where the underlying ROF depends on 2^t variables, for $2^t \leq n$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, over $\mathbb{F}$, $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 34.** *For every field $\mathbb{F}$, there is a hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\log n)}$, that hits every $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

Since a hitting set for all polynomials of the form $g - h$ where $g, h \in \mathcal{C}$ is the same as an interpolating set for $\mathcal{C}$, the following theorem gives an interpolating set for the orbit of ROANFs.

▶ **Theorem 35.** *Let $f_1 = ANF_{\Delta_1}(A_1\boldsymbol{x} + \boldsymbol{b}_1), f_2 = ANF_{\Delta_2}(A_2\boldsymbol{x} + \boldsymbol{b}_2) \in ANF^{GL_n^{aff}(\mathbb{F})}$ and $f = f_1 - f_2$. Set $k \triangleq 2\max\{\Delta_1, \Delta_2\} + 7$ and let $\mathcal{G}$ be any uniform k-independent polynomial map, over $\mathbb{F}$. If $f \neq 0$ then $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 36.** *For any field $\mathbb{F}$, the class $ANF_{\Delta}^{GL_n^{aff}(\mathbb{F})}$, for $4^{\Delta} \leq n$, admits an interpolating set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\Delta)}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \geq n^2$.*

Finally, we observe that the randomized algorithm of Gupta, Kayal And Qiao [35], for reconstructing *random algebraic formula* (for a natural definition of a random formula), yields a randomized reconstruction algorithm for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. Naturally, the reconstruction is up to the symmetry group of ROANFs.

▶ **Theorem 37** (A special case of Theorem 1.1 of [35])**.** *Let T be a finite subset of $\mathbb{C}$. Let $n, \Delta \geq 1$ be integers such that $s \triangleq 4^{\Delta} \leq n$. Given black-box access to the output f of a circuit $\Phi \in ANF^{GL_n^{aff}(\mathbb{C})}$, with probability at least $1 - \frac{n^2 s^{O(1)}}{|T|}$ (on internal randomness), Algorithm 6.9 of [35] successfully computes a tuple of s linearly independent linear functions $L = (\ell_1, \ldots, \ell_s) \in (\mathbb{C}[\boldsymbol{x}])^s$ such that $f = ANF_{\Delta}(\ell_1, \ldots, \ell_s)$, and the ℓ_is are identical to the labels of the leaves of Φ up to $TS_n(\mathbb{C})$-equivalence (see Definition 22). Moreover, the running time of the algorithm is poly$(n, s, \log(|T|))$.*

▶ **Remark 38.** *Theorem 1.1 of [35] is stated only for characteristic zero fields. However, in Remark 6.10 they explain how to make the algorithm work over any characteristic, for a large enough field. Thus, Theorem 37 also holds over large enough fields in arbitrary characteristic.*

▶ **Remark 39.** *As a direct implication of Theorem 35, the reconstruction algorithm of Theorem 37 can be converted into a zero-error algorithm, with expected quasipolynomial running time: Given black-box access to some $f_1 \in ANF^{GL^{aff}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 37 on input f_1, and then verify $f_1 = f_2$ using Corollary 36.*

1.2.3 Dense subclasses of ΣΠΣ

We start by defining the canonical diagonal tensor of degree d and rank s, $\mathrm{T}_{s,d} \in \mathbb{F}[x_{1,1}, \ldots, x_{s,d}]$, and the resulting class of polynomials $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Definition 40.** *Let $T_{s,d} \triangleq \sum_{i=1}^{s} \prod_{j=1}^{d} x_{i,j}$. I.e., it is a sum of s variable-disjoint monomials. For $n \geq s \cdot d$, we denote with $T_{s,d}^{GL_n^{aff}(\mathbb{F})}$ the orbit of $T_{s,d}$ over $\mathbb{F}$, under the action of the affine group. Finally, we denote with $\mathcal{T}^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exist s and d such that $n \geq s \cdot d$ and $f_n(x_1, \ldots, x_n) \in T_{s,d}^{GL_n^{aff}(\mathbb{F})}$.*

Clearly, $\mathrm{T}_{s,d}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \subset \Sigma^{[s]}\Pi^{[d]}\Sigma$. We next define the class consisting of orbits of sparse polynomials.

▶ **Definition 41.** *Let $\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})}$ denote the class of families of polynomials that are computed by orbits of depth-2 circuits, of polynomially bounded size, over $\mathbb{F}$. I.e., it is all families $(f_n)_n$, of polynomially bounded degree, such that for some polynomially bounded $m(n)$, there exist $\Sigma^{m(n)}\Pi^{\deg(f_n)}$ circuits Φ_m, in $k \leq n$, many variables, such that $f_n \in \Phi_m^{GL_n^{\mathit{aff}}(\mathbb{F})}$.*

As before we first give the basic observation connecting all three classes.

▶ **Theorem 42.** *For every field $\mathbb{F}$ it holds that*

$$\mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})} \subsetneq \Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})} \subseteq \Sigma\Pi\Sigma(\mathbb{F}) \ ,$$

and for fields of size $|\mathbb{F}| \geq n+1$

$$\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})} \subsetneq \Sigma\Pi\Sigma(\mathbb{F}) \ .$$

In addition,

$$\overline{\mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})}} = \overline{\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})}} = \overline{\Sigma\Pi\Sigma(\mathbb{F})} \ . \tag{7}$$

Our main results for this section are a quasipolynomial-size hitting set for the class $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, and a polynomial-size interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Theorem 43.** *Let $0 \neq g \in \mathbb{F}[\boldsymbol{x}]$ have sparsity $\leq 2^t$. Let $(A, \boldsymbol{b}) \in GL_n^{\mathit{aff}}(\mathbb{F})$, and $f(\boldsymbol{x}) = g(A\boldsymbol{x} + \boldsymbol{b})$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, $f \circ \mathcal{G} \neq 0$.*

▶ **Corollary 44.** *For any integers s, d, n, there exists an explicit hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = (nd)^{O(\log s)}$, such that $\mathcal{H}$ hits every nonzero polynomial $f \in \left(\Sigma^{[s]}\Pi^{[d]}\right)^{GL_n^{\mathit{aff}}(\mathbb{F})}$. If $|\mathbb{F}| \leq n \cdot d$ then we let $\mathcal{H}$ be defined over an extension field $\mathbb{K}$ of $\mathbb{F}$ of size $|\mathbb{K}| > n \cdot d$.*

We next state our result concerning an interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

▶ **Theorem 45.** *Let $n, s_1, s_2, d_1, d_2 \in \mathbb{N}$ be such that $n \geq s_1 \cdot d_1, s_2 \cdot d_2$. For $i \in \{1, 2\}$ let $f_i \in T_{s_i,d_i}^{GL_n(\mathbb{F})}$, and let $f = f_1 - f_2$. If $f \neq 0$, then any uniform 6-independent polynomial map $\mathcal{G}$ satisfies $f \circ \mathcal{G} \neq 0$.*

Finally we note that the randomized reconstruction algorithm of Kayal and Saha [45], which works for (as it is termed in their paper) "non-degenerate" homogeneous depth-3 circuits, works for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. This follows from the observation that $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ circuits are always non-degenerate.

▶ **Theorem 46** (special case of Theorem 1 of [45])**.** *Let $n, d, s \in \mathbb{N}$, $n \geq (3d)^2$ and $s \leq (\frac{n}{3d})^{\frac{d}{3}}$. Let $\mathbb{F}$ be a field of characteristic zero or greater than ds^2. There is a randomized $\mathit{poly}(n, d, s) = \mathit{poly}(n, s)$ time algorithm which takes as input black-box access to a polynomial f that is computable by a $T_{s,d}^{GL_n^{\mathit{aff}}(\mathbb{F})}$ circuit, and outputs a $T_{s,d}^{GL_n^{\mathit{aff}}(\mathbb{F})}$ circuit Φ computing f with high probability. Furthermore, Φ is unique up to $TPS_{s,d}(\mathbb{F})$-equivalence (see Definition 25).*

▶ **Remark 47.** *As in Remark 39, Theorem 45 enables us to convert the reconstruction algorithm of Theorem 46 to a zero-error algorithm, with expected polynomial running time. Given black-box access to some $f_1 \in \mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 46 on input f_1, and then verify $f_1 \equiv f_2$ by applying Theorem 45 to $f = f_1 - f_2$.*

1.2.4 Robust hitting sets?

As we showed in Observation 14, if a hitting set $\mathcal{H}$ for a circuit class $\mathcal{C}$ is *robust*, then $\mathcal{H}$ hits $\overline{\mathcal{C}}$ as well. It is thus natural to ask whether our interpolating sets are already robust. Our next result shows that the property of being a t-independent map, which was sufficient for the constructions in Theorems 28, 30, 33, 35, 43, and 45 (for the appropriate values of t), by itself is not sufficient for obtaining robust hitting sets. We prove this by constructing an independent polynomial map which gives rise to a provably non-robust hitting set. Our construction is the same as the one given by Forbes et al. [27] (Construction 6.3 in the full version).

▶ **Theorem 48.** *Let $\mathbb{F}$ be of characteristic zero. For every t, there exists a uniform t-independent polynomial map $\mathcal{G}$ and a nonzero polynomial f such that $f \circ \mathcal{G} \equiv 0$, and f can be computed by a $\Sigma\Pi\Sigma$ formula of size $t^{O(\sqrt{t})}$. If $\mathbb{F}$ has a positive characteristic then f can be computed by a $\Sigma\Pi\Sigma$ formula of size t^t, or by a general formula of size $t^{O(\log t)}$. Furthermore, for a certain arrangement of the variables in a $\sqrt{n} \times \sqrt{n}$ matrix, f can be taken to be the determinant of any $(t+1) \times (t+1)$ minor.*

1.3 Polynomial Identity Testing

So far we discussed our work from the perspective of dense subclasses of classes for which no strong lower bounds are known. Here we put our work in the context of the polynomial identity testing problem.

Polynomial Identity Testing (PIT for short) is the problem of designing efficient deterministic algorithms for deciding whether a given arithmetic circuit computes the identically zero polynomial. PIT has many applications, e.g. deciding primality [1], finding a perfect matching in parallel [23, 69] etc., and strong connection to circuit lower bounds [38, 22, 19, 32]. See [67, 62, 63] for surveys on PIT and [50] for a survey of algebraic hardness-randomness tradeoffs.

PIT is considered both in the white-box model, in which we get access to the graph of computation of the circuit, and in the black-box model in which we only get query access to the polynomial computed by the circuit. Clearly, a deterministic PIT algorithm in the black-box model is equivalent to a hitting set for the circuit class. In this work we only focus on the black-box model.

The continuant polynomial and algebraic branching programs

The continuant polynomial is trivially computed by width-2 *Algebraic Branching Programs* (ABPs). Recall that an ABP of depth-d and width-w computes polynomials of the form $\mathrm{Trace}\left(M_1(\boldsymbol{x}) \cdot \ldots \cdot M_d(\boldsymbol{x})\right)$, where each M_i is a $w \times w$ matrix whose entries contain variables or field elements. Ben-Or and Cleve proved that every polynomial in $\mathrm{VP_e}$ can be computed by a width-3 ABP of polynomial-size [8].

Raz and Shpilka gave the first polynomial-time white-box PIT algorithm for read-once ABPs (ABPs in which every variable can appear in at most one matrix) [58]. Forbes, Saptharishi and Shpilka gave the first quasipolynomial-sized hitting set for read-once ABPs (ROABPs) [25]. This result was slightly improved in [31] for the case where the width of the ROABP is small. Anderson et al. gave a subexponential hitting set for read-k ABPs [5]. We note that none of these models is strong enough to contain the orbit $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. For ABPs that are not constant-read we do not have sub-exponential time PIT algorithms. Thus, the following is an interesting open problem (recall that by the result of Ben-Or and Cleve a PIT algorithm for width-3 ABPs works for $\mathrm{VP_e}$ as well).

▶ **Problem 49.** *Give a sub-exponential time PIT algorithm for ABPs of width-2.*

It is interesting to note that by a result of Saha, Saptharishi and Saxena [59], PIT for ABPs of width-2 would yield PIT for $\Sigma\Pi\Sigma$ circuits.

Although we do not have a PIT algorithm for general branching programs, in [44] Kayal et al. gave an average-case reconstruction algorithm for low width ABPs. Kayal, Nair and Saha obtained a significantly better algorithm in [43]. Their algorithm succeeds w.h.p, provided the ABP satisfies four non-degeneracy conditions (these conditions are defined in Section 4.3 of [43]). However, the ABP computing the continuant polynomial does not satisfy the non-degeneracy conditions that are required for their algorithm to work. Thus, Theorem 31 does not follow from [43].

To the best of our knowledge, $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ is the first natural[9] computational class that is dense in $\mathrm{VP_e}$ for which a polynomial (or even sub-exponential)-sized interpolating set (or a hitting set) is known.

Read-Once formulas

Hitting sets for read-once formulas were first constructed by Volkovich and Shpilka [66], who gave quasipolynomial-sized hitting set for the model, as well as a deterministic reconstruction algorithm of the same running time (earlier randomized reconstruction algorithms were known [16, 15]). Minahan and Volkovich obtained a polynomial-sized hitting set for the class, which led to a similar improvement in the running time of the reconstruction algorithm [54]. Anderson, van Melkebeek and Volkovich constructed a hitting set of size $n^{k^{O(k)}+O(k\log n)}$ for read-k formulas [6]. All these results work in a slightly stronger model in which we allow to label leaves with univariate polynomials, of polynomial degree, such that every variable appears in at most one polynomial, or with sparse polynomials on disjoint sets of variables.

The read-once models that we consider here, $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$, can be viewed as read-once formulas composed with a layer of addition gates with the restriction that the bottom layer of additions computes linearly independent linear functions. We note that these models do not fall into any of the previously studied models, as a variable can appear in all the linear functions.

As is the case with $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, our hitting sets for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ are the first sub-exponential-sized hitting sets for natural dense subclasses of $\mathrm{VP_e}$.

Small depth circuits

The class of $\Sigma\Pi$ circuits was considered in many works, see e.g. [9, 48] and polynomial-sized hitting sets were constructed. The class of $\Sigma\Pi\Sigma$ circuits also received a lot of attention but with lesser success. Dvir and Shpilka [21] and Karnin and Shpilka [40] gave the first quasipolynomial-time white-box and black-box PIT algorithms for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, respectively. Currently, the best result is by Saxena and Seshadhri who gave a hitting set of size $(nd)^{O(k)}$ for such circuits [64]. In [20] a subexponential-size hitting set for *multilinear* $\Sigma\Pi\Sigma$ circuits was given. In [2], Agrawal et al. gave a hitting set of size $n^{O(1)}\cdot(kd)^{O(r)}$ for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, where r is an upper bound on the *algebraic rank* of the multiplication gates in the circuit. Thus, known quasipolynomial-size hitting sets for subclasses of $\Sigma\Pi\Sigma$ circuits are known when the fan-in of the top gate is poly-logarithmic, or when the algebraic rank of

[9] It is hard to define what a natural class means, but, for example the set of all polynomials in $\mathrm{VP_e}$ with a nonzero free term has a trivial hitting set, but is not a "computational" subclass.

the set of multiplication gates is poly-logarithmic. In contrast, polynomials in $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, when viewed as $\Sigma\Pi\Sigma$ circuits, can have polynomially many multiplication gates and their algebraic rank can be n. On the other hand, the corresponding $\Sigma\Pi\Sigma$ circuits are such that the *different* linear functions that are computed at their bottom layer are linearly independent (when we view linear functions that are a constant multiple of each other as the same function). Thus, our Corollary 44 provides a hitting set for a new subclass of $\Sigma\Pi\Sigma$ circuits.

To the best of our knowledge, our results for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ give the first sub-exponential size hitting sets for natural subclasses that are dense in $\Sigma\Pi\Sigma$.

1.4 More related work

Approximations in algebraic complexity were first studied by Bini et al. in the context of algorithms for matrix multiplication [12]. For more on the history of border rank in the context of matrix multiplication see notes of chapter 15 in [18]. More recently, influenced by the GCT program, a lot of research was invested in trying to find polynomials characterizing tensors of small rank. See [51] for a discussion on this approach. More recently, Kumar proved that *every* polynomial over $\mathbb{C}$ can be approximated by a $\Sigma^{[2]}\Pi\Sigma$ circuit (of exponential degree) [49].

Very little is known about the closure of circuit classes. Forbes observed that the class of ROABPs is closed [24]. I.e. ROABP $= \overline{\text{ROABP}}$. We are not aware of other collapses or separation between general "natural" classes and their closures.

Beside the reconstruction algorithms mentioned earlier, reconstruction algorithms are known for $\Sigma\Pi$ circuits [9, 48]; for random depth three *powering* circuits [42]; for set-multilinear $\Sigma\Pi\Sigma$ and ROABPs [7, 47]; for $\Sigma\Pi\Sigma$ circuits with bounded top fan-in [65, 41, 68]; and for multilinear depth-4 circuits with a constant top fan-in [34, 11].

In general, we do not expect the reconstruction problem to be solvable efficiently, as the problem of finding the minimal circuit computing a given polynomial is a notoriously hard problem. A detailed discussion on the hardness of reconstruction can be found in [43].

Independently and concurrently with our work Saha and Thankkey gave PIT algorithms for orbits of different models of read-once oblivious algebraic branching programs (ROABPs) and for constant-depth, constant-occur formulas [60]. Their results concerning ROABPs were recently improved by Bhargava and Ghosh [10]. Interestingly, both [60, 10] use k-wise independent maps in their construction. We note that the only model that is studied in this paper and in [60, 10] is that of (orbits of) sparse polynomials. For orbits of sparse polynomials are hitting set is potentially much smaller than those constructed in [60, 10] as it does not depend on the individual degrees appearing in the sparse polynomial.

Simultaneously and independently, Saha and Thankey [60] studied PIT for orbits of related computational models. Specifically, they obtained quasi-polynomial sized hitting sets for: Low-individual-degree polynomials computable by commutative ROABP; Multilinear polynomials computable by constant-width ROABP; Polynomials computable by constant-depth, constant-occur formulas with low-individual-degree sparse polynomials at the leaves; and Polynomials computable by occur-once formulas with low-individual-degree sparse polynomials at the leaves. We refer the reader for their paper for definitions of these models. The results of [60] are mostly disjoint from ours, except for the model of sparse polynomials that is captured by commutative ABPs. In this case our result is superior to theirs as their hitting set has an exponential dependence in the individual degrees, while ours work for any polynomial degree sparse polynomial. It is interesting to note that the hitting set constructions of [60] are also based on k-independent maps.

1.5 Proof technique

Our proofs are based on the following simple yet important, and as far as we know novel, observations concerning k-independent polynomial maps. Specifically, our proofs are based on the following two claims:

1. If we have a hitting-set generator H for nonzero polynomials of the form $\frac{\partial f}{\partial x_1}$, for $f \in \mathcal{C}$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This is proved in Lemma 61.
2. Similarly, we prove that if we have a hitting-set generator H for nonzero polynomials of the form $f\big|_{\ell=0}(A\boldsymbol{x}+\boldsymbol{b})$, for $f \in \mathcal{C}$, a linear function ℓ, and an invertible affine transformation $(A, \boldsymbol{b})$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This follows from Lemma 62.

By applying these claims $k + r$ times we get that composition with a $(k + r)$-independent map allows to reduce the problem of hitting a class $\mathcal{C}$ to hitting polynomials of the form $\frac{\partial^k f}{\partial x_{i_1}\partial x_{i_2}\cdots\partial x_{i_k}}\Big|_{\ell_1=\ldots=\ell_r=0}$. Thus, if we could prove that for a class $\mathcal{C}$, there is such a sequence of derivatives and restrictions that simplifies the polynomials in it to a degree that they can be easily hit by some map H, then we conclude that $H + \mathcal{G}_{k+r}$, for a $(k + r)$-independent map $\mathcal{G}_{k+r}$, is a hitting set generator for $\mathcal{C}$.

It seems that all that is left to do is prove that for each of the orbits that we consider in Section 1.2 that is such small k and r. However, a potential problem is that a partial derivative of the polynomial $g(\boldsymbol{x}) = f(A\boldsymbol{x} + \boldsymbol{b})$ gives $\frac{\partial g}{\partial x_1} = \sum_{i=1}^{n} \frac{\partial f}{\partial y_i} \cdot \frac{\partial \ell_i}{\partial x_1}$, where ℓ_i is the ith coordinate of $A\boldsymbol{x} + \boldsymbol{b}$. Thus, it is no longer a derivative composed with an affine transformation but rather a sum of such derivatives, which could lead to polynomials outside of our class. For example, it is not hard ot prove that if we compose the ROF $y_1 \cdot y_2 \cdot y_3$ with $(x_1, x_1 + x_2, x_1 + x_3)$ and then take a derivative according to x_1, then the resulting polynomial, $\frac{\partial(x_1\cdot(x_1+x_2)\cdot(x_1+x_3))}{\partial x_1} = 3x_1^2 + 2x_1 \cdot (x_2 + x_3) + x_2 \cdot x_3$, is not in the orbit of any ROF. The solution to this problem is to take a *directional derivative* in a direction coming from a *dual basis*. For example if $\ell_i(\boldsymbol{v}_j) = \delta_{i,j}$ then $\frac{\partial g}{\partial \boldsymbol{v}_1} = \frac{\partial f}{\partial x_1}(A\boldsymbol{x} + \boldsymbol{b})$ (see Lemma 60). Now, comes another important observation: If H is a hitting-set generator for nonzero polynomials of the form $\frac{\partial f}{\partial \boldsymbol{v}}$, for $f \in \mathcal{C}$ and a direction $\boldsymbol{v}$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. The point is that if $\frac{\partial f}{\partial \boldsymbol{v}} \circ H \neq 0$ then for some i, $\frac{\partial f}{\partial x_i} \circ H \neq 0$ and the claim follows from the first claim above. Thus, composition with $(k + r)$-independent maps allows us to reduce the problem of hitting a class $\mathcal{C}$ to finding a generator for polynomials that are obtained as a restriction to a subspace of co-dimension r of a directional partial derivative of order k of polynomials in $\mathcal{C}$.

Let us demonstrate this idea for the case of orbits of sparse polynomials. I.e. to polynomials of the form $g(\boldsymbol{x}) = f(A\boldsymbol{x} + \boldsymbol{b})$, where the number of monomials in f is at most 2^t. It is not hard to see that there is a variable x_i such that if we consider $f\big|_{x_i=0}$ and $\frac{\partial f}{\partial x_i}$ then one of these polynomials has at most 2^{t-1} monomials.[10] Thus, after a a sequence of at most t partial derivatives and restrictions, we get to a polynomial with only one monomial that we can easily hit. Hence after at most t directional derivatives and restrictions to a subspace, we get that g is a product of linear forms, which we can easily hit. This proves that any $(t + 1)$-independent map hits such nonzero polynomials g.

[10] This is not exactly accurate – it only holds if f is not divisible by some variable x_i. However, the case where there is a monomial dividing f is also quite easy to handle as it is enough to hit the polynomial obtained after dividing by that monomial (since a composition with a 1-independent map keeps any nonzero linear function nonzero).

To obtain interpolating sets for our classes (and also a reconstruction algorithm for the orbit of the continuant polynomial), we prove that if two polynomials in the orbit, of any of the classes that we consider, are different, then there is a sequence of a few (directional) partial derivatives and restrictions that makes one of them zero while keeping the other nonzero. Using this and the ideas from above we construct our interpolating sets.

▶ **Remark 50.** *In this version of the paper we only give proofs of the main properties of k-wise independent maps (outlined above), as these are the main tool that we used in all our proofs. The full version can be found at [53].*

1.6 Discussion

As Theorem 48 shows, our hitting sets are not necessarily robust. It is thus an outstanding open problem to find a way to convert a hitting set to a robust one (recall Problem 1).

The following toy example demonstrates that converting a hitting set for a class $\mathcal{C}$ to a robust hitting set for $\mathcal{C}$, cannot be done in a black-box manner and one has to use information about $\mathcal{C}$ for that: let $\mathcal{C}(\mathbb{F})$ be the class of all polynomials with non-zero free term. A trivial hitting set for $\mathcal{C}$ would simply be the singleton set $\mathcal{H} = \{\mathbf{0}\}$. On the other hand, it is clear that $\overline{\mathcal{C}} = \mathbb{F}[\boldsymbol{x}]$, so making $\mathcal{H}$ robust would yield a hitting set for *all* polynomials. Note, however, that this is not a "computational class."

Another potential approach for obtaining robust hitting sets follows from the observation that the set of queries made by a non-adaptive deterministic black-box reconstruction algorithm, $\mathcal{A}$, for $\mathcal{C}$, which is *continuous* at 0 (i.e. at the identically zero polynomial) is a robust hitting set for $\mathcal{C}$. The reason is, that if $0 \neq f \in \overline{\mathcal{C}}$ and $\{f_k\}_{k=1}^{\infty} \subseteq \mathcal{C}$ converges to f, then for large enough k: $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$. As the f_k sequence converges and polynomial evaluation is continuous (and their evaluation vectors are bounded), the sequence $\boldsymbol{v}_k = f_k\big|_{\mathcal{H}} \subseteq \mathbb{C}^{|\mathcal{H}|}$ must also converge to some vector $\boldsymbol{v} = f\big|_{\mathcal{H}} \in \mathbb{C}^{|\mathcal{H}|}$. If $\boldsymbol{v} = \mathbf{0}$ then the continuity of $\mathcal{A}$ at $\mathbf{0}$ implies the coefficients of the polynomials $f_k(\boldsymbol{x})$ must also converge to zero, as $\mathcal{A}(\mathbf{0}) = 0$. This would contradict $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$ for large enough k, so $\boldsymbol{v} \neq \mathbf{0}$ and thus $\mathcal{H}$ hits $\overline{\mathcal{C}}$.

Thus, an interesting challenge is to derandomize the reconstruction algorithms given in Theorems 31, 37, and 46, hoping that the resulting algorithms are continuous at $\mathbf{0}$. We note however, that currently we do not even have efficient deterministic root-finding algorithms over $\mathbb{C}$. It is also known that in general, finding the minimal circuit for a polynomial can be very difficult. E.g., in [36, 70] it was shown that the question of computing, or even approximating, tensor rank, for degree 3 tensors, is NP hard, over any field.

▶ Remark 51. In Theorem 45, we have seen that any uniform $O(\log(sn))$-independent polynomial map $\mathcal{G}$ is an interpolating set generator for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$; i.e, $\mathcal{G}$ induces an interpolating set $\mathcal{H}$ for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. On the other hand, in Theorem 48, we constructed such a map $\mathcal{G}$, with the additional property that $\mathcal{G}$ is *not* a hitting set generator for $\Sigma\Pi\Sigma$ circuits. In particular, this implies that the induced (non-efficient) reconstruction map $\mathcal{A}$ (that takes $f(\mathcal{H})$ and returns a circuit computing f) is not continuous at $\mathbf{0}$.

We conclude this section with a somewhat vague question.

▶ **Problem 52.** *Find a "computational" class of polynomials $\mathcal{C}$ with a known hitting set $\mathcal{H}$, such that $\overline{\mathcal{C}} \neq \mathcal{C}$, and convert $\mathcal{H}$ to a robust hitting set.*

We note that the closure of $\Sigma\bigwedge\Sigma$ circuits (i.e. circuits computing polynomials of the form $\sum_i \ell_i(\boldsymbol{x})^d$, for linear functions ℓ_i) is contained in the class of commutative read-once algebraic branching programs (see [25]). Thus, the hitting set for the latter class gives a robust hitting set for the former [25]. However, we seek an example in which there is an "interesting" conversion of a hitting set to a robust one.

2 k-independent polynomial maps and their properties

▶ **Observation 53.** *It holds that*

1. *If $\mathcal{G}(\boldsymbol{y},\boldsymbol{z})$ is a $(k+1)$-independent polynomial map, then there exists a subset of variables S and an assignment $\boldsymbol{\alpha}\in\mathbb{F}^{|S|}$ such that $\mathcal{G}\big|_{S=\boldsymbol{\alpha}}$ is a k-independent polynomial map.*
2. *For any $k\geq 1$, the n coordinates of any k-independent polynomial map are $\mathbb{F}$-linearly independent.*
3. *Let $\ell_1(\boldsymbol{x})$ and $\ell_2(\boldsymbol{x})$ be linearly independent linear functions in $\mathbb{F}[\boldsymbol{x}]$. Let $\mathcal{G}(\mathbf{y},z_1,z_2)$ be any 2-independent polynomial map. Consider $\ell_1\circ\mathcal{G}$ and $\ell_2\circ\mathcal{G}$ as polynomials in z_1,z_2 over $\mathbb{F}(\mathbf{y})$. Then, $(\ell_1\circ\mathcal{G})^{[1]}$ and $(\ell_2\circ\mathcal{G})^{[1]}$ are linearly independent, as linear forms in z_1,z_2 over $\mathbb{F}(\mathbf{y})$.*

We next give the construction of [66] of a k-independent polynomial map (denoted G_k in [66]).

▶ **Definition 54.** *Fix n and a set of n distinct field elements $\mathcal{A}=\{\alpha_1,\ldots,\alpha_n\}\subseteq\mathbb{F}$.*[11] *For every $i\in[n]$ let $L_i(w):\mathbb{F}\to\mathbb{F}$ be the ith Lagrange Interpolation polynomial for the set $\mathcal{A}$. That is, each $L_i(w)$ is polynomial of degree $n-1$ that satisfies $L_i(\alpha_j)=\delta_{i,j}$. We define $\mathcal{G}_1^{SV}(y_1,z_1):\mathbb{F}^2\to\mathbb{F}^n$ as:*

$$\mathcal{G}_1^{SV}(y_1,z_1)\triangleq\left(L_1(y_1)\cdot z_1,L_2(y_1)\cdot z_1,\ldots,L_n(y_1)\cdot z_1\right),$$

and for any $k\geq 1$, we define $\mathcal{G}_k^{SV}:\mathbb{F}^{2k}\to\mathbb{F}^n$ as:

$$\begin{aligned}\mathcal{G}_k^{SV}(\boldsymbol{y},\boldsymbol{z})&\triangleq\mathcal{G}_1^{SV}(y_1,z_1)+\mathcal{G}_1^{SV}(y_2,z_2)+\ldots+\mathcal{G}_1^{SV}(y_k,z_k)\\&=\left(\sum_{j=1}^{k}L_1(y_j)\cdot z_j,\sum_{j=1}^{k}L_2(y_j)\cdot z_j,\ldots,\sum_{j=1}^{k}L_n(y_j)\cdot z_j\right).\end{aligned}$$

▶ **Observation 55.** *$\mathcal{G}_k^{SV}$ is a k-independent polynomial map, in which each variable has degree at most $n-1$.*

The generator $\mathcal{G}_k^{\mathrm{SV}}$ can be converted to a uniform k-independent polynomial map by adding another k control variables $y_{k+1},\ldots,y_{2k}$, and swapping out the $L_i(y_j)$s for their homogenizations $y_{j+k}^{n-1}L_i\left(\frac{y_j}{y_{j+k}}\right)$:

▶ **Definition 56.** *With the notation used in Definition 54, define the* uniform SV-generator *with k independence $\mathcal{G}_k^{SV\text{-}hom}:\mathbb{F}^{3k}\to\mathbb{F}^n$ as:*

$$\begin{aligned}&\mathcal{G}_k^{SV\text{-}hom}(y_1,\ldots,y_{2k},z_1,\ldots,z_k)\\&\quad\triangleq y_{1+k}^{n-1}\cdot\mathcal{G}_1^{SV}\left(\frac{y_1}{y_{1+k}},z_1\right)+y_{2+k}^{n-1}\cdot\mathcal{G}_1^{SV}\left(\frac{y_2}{y_{2+k}},z_2\right)+\ldots+y_{2k}^{n-1}\cdot\mathcal{G}_1^{SV}\left(\frac{y_k}{y_{2k}},z_k\right)\\&\quad=\left(\sum_{j=1}^{k}y_{j+k}^{n-1}L_1\left(\frac{y_j}{y_{j+k}}\right)\cdot z_j,\sum_{j=1}^{k}y_{j+k}^{n-1}L_2\left(\frac{y_j}{y_{j+k}}\right)\cdot z_j,\ldots,\sum_{j=1}^{k}y_{j+k}^{n-1}L_n\left(\frac{y_j}{y_{j+k}}\right)\cdot z_j\right).\end{aligned}$$

▶ **Observation 57.** *$\mathcal{G}_k^{SV\text{-}hom}$ is a uniform k-independent polynomial map, with individual degrees at most $n-1$.*

[11] If $|\mathbb{F}|<n$ then we take these elements from an appropriate extension field of $\mathbb{F}$.

We next show how we can use k-independent polynomial maps in order to, roughly, simulate a kth order directional derivative or, project a polynomial to a subspace of co-dimension k. We first need to define the notion of a directional derivative.

▶ **Definition 58.** *For an n-variate polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}^n$, the* derivative of $f(\boldsymbol{x})$ in the direction $\boldsymbol{v}$ *is defined as:*

$$\frac{\partial f}{\partial \boldsymbol{v}} = \sum_{i=1}^{n} v_i \cdot \frac{\partial f}{\partial x_i}.$$

If $\mathbb{F}$ has positive characteristic then by $\frac{\partial F}{\partial x_i}$ we refer to the formal derivative (which in the case of fields of characteristic zero is equal to the analytical definition). Observe that we still have that

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial (fg)}{\partial x} = \frac{\partial f}{\partial x} \cdot g + \frac{\partial g}{\partial x} \cdot f$$

and

$$\frac{\partial f\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right)}{\partial x_k} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i}\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right) \cdot \frac{\partial g_i}{\partial x_k}\ ,$$

where in the last expression f is an m variate polynomial, and $g_1, \ldots, g_m$ are n variate polynomials.

We shall often take derivatives according to a *dual set* to a set of linearly independent linear functions:

▶ **Definition 59.** *A dual set for m linearly independent linear functions (recall that we say that linear functions are linearly independent if and only if their degree-1 homogeneous parts are linearly independent) in $n \geq m$ variables, $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$ is a set of m vectors $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ such that $\ell_i^{[1]}(\boldsymbol{v}_j) = \delta_{i,j}$.*

▶ **Lemma 60.** *Let $\ell_1, \ldots, \ell_m \in \mathbb{F}[x_1, \ldots, x_n]$, for $n \geq m$, be linearly independent linear functions. Let $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ be a dual set. Let $g \in \mathbb{F}[y_1, \ldots, y_m]$ be a polynomial. Then, for $f(\boldsymbol{x}) = g\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)$ it holds that*

$$\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)\ .$$

Proof.

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) &= \sum_j v_{i,j} \cdot \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = \sum_{j,k} v_{i,j} \cdot \frac{\partial \ell_k}{\partial x_j} \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) \\ &= \sum_k \ell_k^{[1]}(\boldsymbol{v}_i) \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)\ .\end{aligned}$$

◀

▶ **Lemma 61.** *Let $f \in \mathbb{F}[\boldsymbol{x}]$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$. Let $H(\boldsymbol{w}) : \mathbb{F}^t \to \mathbb{F}^n$ be a polynomial map in variables $\boldsymbol{w}$, and let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map such that $\mathrm{var}(H) \cap \mathrm{var}(\mathcal{G}) = \emptyset$. Then, for any $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^n$:*

$$\frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k} \circ H \neq 0 \quad \Rightarrow \quad f \circ (\mathcal{G} + H) \neq 0\ .$$

Proof. By definition of k-independent polynomial maps, $\mathcal{G} = \mathcal{G}_1(\boldsymbol{y_1}, z_1) + \ldots + \mathcal{G}_k(\boldsymbol{y_k}, z_k)$ for some variable-disjoint 1-independent polynomial maps $\mathcal{G}_1, \ldots, \mathcal{G}_k$. It is therefore enough to prove the lemma for $k = 1$, as we can replace f with $\frac{\partial^{k-1} f}{\partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k}$, H with $H + \mathcal{G}_2 + \ldots + \mathcal{G}_k$ and $\mathcal{G}$ with $\mathcal{G}_1$; by iterative application of the result for $k = 1$, we will get the general result for an arbitrary $k \in \mathbb{N}$.

Denote $H = (H_1, H_2, \ldots, H_n)$. By Definition 58, the condition $\frac{\partial f}{\partial \boldsymbol{y}} \circ H \neq 0$ implies that there exists some $i \in [n]$ such that $\frac{\partial f}{\partial x_i} \circ H \neq 0$. Assume, WLOG, $\frac{\partial f}{\partial x_1} \circ H \neq 0$. As $\mathcal{G}$ is a 1-independent polynomial map, there exists some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y_1}|}$ such that $f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}} = f(z_1 + H_1, H_2, \ldots, H_n)$; denote $g \triangleq f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}}$. As no coordinate of H depends on z_1:

$$\frac{\partial g}{\partial z_1} = \frac{\partial (z_1 + H_1)}{\partial z_1} \cdot \frac{\partial f}{\partial x_1}(z_1 + H_1, H_2, \ldots, H_n) = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(z_1 + H_1, H_2, \ldots, H_n)$$

and therefore:

$$\left.\frac{\partial g}{\partial z_1}\right|_{z_1 = 0} = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(0 + H_1, H_2, \ldots, H_n) = \left(\frac{\partial f}{\partial x_1}\right) \circ H \neq 0 \,.$$

As g is a projection of $f \circ (\mathcal{G} + H)$, it follows that $f \circ (\mathcal{G} + H) \neq 0$. ◀

The next lemma shows how to use k-independent maps in order to project a polynomial to a subset of its coordinates.

▶ **Lemma 62.** *Let $m \leq n \in \mathbb{N}$ and $g(\boldsymbol{w}) \in \mathbb{F}[w_1, \ldots, w_m]$. Let $f(\boldsymbol{x}) = g(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x}))$ for linearly independent linear functions $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$. Let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map. For a set $S \subseteq [n]$ of size k denote by $\tilde{g}(x_i : i \in [m] \setminus S) = g\big|_{S=0}$ the projection of g to the variables outside of S. Then, there exist linearly independent linear functions $\{\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S\}$, additional linear functions $\boldsymbol{L}(\boldsymbol{x}) = (L_1(\boldsymbol{x}), \ldots, L_k(\boldsymbol{x}))$ and an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that:*

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{L}(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S) \,.$$

Proof. It is enough to prove the lemma for the case $k = 1$, as we may then define $\tilde{f}(\boldsymbol{x}) \triangleq f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L_1(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_1(\boldsymbol{x}), \ldots, \tilde{\ell}_{m-1}(\boldsymbol{x}))$ and apply the result iteratively. Thus, assume $k = 1$, and WLOG assume $S = \{x_1\}$ (thus, $\tilde{g}(w_2, \ldots, w_m) = g(0, w_2, \ldots, w_m)$).

Let x_i be some variable with a non-zero coefficient in $\ell_1(\boldsymbol{x})$. Such a variable exists as the ℓ_js are linearly independent. For $j \in [m]$, denote $\beta_j = \frac{\partial \ell_j}{\partial x_i}$, i.e. β_j is the coefficient of x_i in ℓ_j. By our choice of i, $\beta_1 \neq 0$. Choose some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that $\mathcal{G}(\boldsymbol{\alpha}, z_1)$ has z_1 in the ith coordinate, and 0 in all other coordinates. Define $L(\boldsymbol{x}) \triangleq -\frac{\ell_1(\boldsymbol{x})}{\beta_1}$, so we get:

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})) = f\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right).$$

Observe that for every i,

$$\ell_i\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})\right) = \ell_i\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right) = \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

In particular, $\ell_1\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})\right) = 0$. For $i = 2, \ldots, m$, define:

$$\tilde{\ell}_i(\boldsymbol{x}) \triangleq \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

As $\ell_1, \ldots, \ell_m$ are linearly independent, it follows that $\tilde{\ell}_2, \ldots, \tilde{\ell}_m$ are also linearly independent. We get that

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x}))) = g(0, \tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) = \tilde{g}(\tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) \,.$$ ◀

2.1 Proof of Theorem 48

We next prove that there are k-independent maps that are provably not robust. The proof is by giving a different construction of such maps that, for an appropriate arrangement of the n variables in a matrix, is guaranteed to output matrices of rank at most k. Thus, a determinant of any $(k+1)\times(k+1)$ minor, a polynomial that has small formulas for small values of k, vanishes on the output of any such map.

The fact that such a construction exists was already noticed in [27] (Construction 6.3 of the full version of the paper). For completeness we repeat the construction here.

Proof. *(of Theorem 48)* Fix the number of variables n and assume WLOG n is a perfect square, i.e., $n=m^2$. We index the variables as $x_{i,j}$ for $i,j\in[m]$. We let $f=\mathrm{Det}_{t+1}$. By [33], over fields of characteristic zero, f has a $t^{O(\sqrt{t})}=O(n)$ sized $\Sigma\Pi\Sigma$ formula, which is polynomial in n for $t=O\left((\log n/\log\log n)^2\right)$. Over fields of positive characteristic the formula size is quasipolynomial in t, and the $\Sigma\Pi\Sigma$ complexity is at most $t!$, which is polynomial in n for $t=O\left(\log n/\log\log n\right)$.

Denote by $\boldsymbol{M}$ the $(t+1)\times(t+1)$ symbolic matrix of variables $\boldsymbol{M}_{i,j}=x_{i,j}$. We first construct a uniform 1-independent polynomial map $\mathcal{G}_1$ such that $\boldsymbol{M}\circ\mathcal{G}_1$ is of rank 1, and define $\mathcal{G}$ to be a sum of t variable-disjoint copies of $\mathcal{G}_1$. As $rank(\boldsymbol{M}\circ\mathcal{G}_1)=1$, we have $rank(\boldsymbol{M}\circ\mathcal{G})\le t$ so $\mathrm{Det}_{t+1}(\boldsymbol{M}\circ\mathcal{G})=0$, as required. We now focus on $\mathcal{G}_1$.

Fix n distinct field elements $\{\alpha_{i,j}\}_{i,j=1}^{m}\subseteq\mathbb{F}$ and let w,y,z be new variables. Define two vectors of polynomials of degree $n-1$, $R=(R_1,\ldots,R_m), C=(C_1,\ldots,C_m)\in\mathbb{F}[y]^m$, such that for every $k\in[m]$ R_k and C_k satisfy

$$R_k(\alpha_{i,j})=\delta_{i,k}\quad\text{and}\quad C_k(\alpha_{i,j})=\delta_{j,k}.$$

Define $\mathcal{G}_1(w,y,z)$ as the $m\times m$ matrix $z\cdot(w^{2n-2}R(\frac{y}{w})\cdot C(\frac{y}{w})^T)$ (the (i,j) entry of $\mathcal{G}_1$ is $z\cdot w^{2n-2}\cdot R_i(\frac{y}{w})\cdot C_j(\frac{y}{w})$). As every coordinate of $\mathcal{G}_1$ is a homogeneous polynomial of degree $2n-1$, $\mathcal{G}_1$ is a uniform polynomial map. For any $i,j\in[m]$ we have that

$$\mathcal{G}_1(1,\alpha_{i,j},z)=z\cdot(R_{i'}(\alpha_{i,j})\cdot C_{j'}(\alpha_{i,j}))_{i',j'\in[m]}=z\cdot(\delta_{i,i'}\delta_{j,j'})_{i',j'\in[m]}\ .$$

The above matrix has z in entry (i,j) and 0 everywhere else, so $\mathcal{G}_1$ is a uniform 1-independent polynomial map. The resulting matrix $\boldsymbol{M}\circ\mathcal{G}_1$ is of rank 1 since it is a product of vectors $R\cdot C^T$, so the variable-disjoint sum $\mathcal{G}=\sum_1^t\mathcal{G}_1(w_i,y_i,z_i)$ is a uniform t-independent polynomial map satisfying $f\circ\mathcal{G}=0$. ◀

References

1 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Ann. of Math*, 2:781–793, 2002.

2 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM J. Comput.*, 45(4):1533–1562, 2016. doi:10.1137/130910725.

3 A. Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht.* PhD thesis, Universität Zürich, Philosophische Fakultät II, 1984.

4 Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Computational Complexity*, 25(1):217–253, 2016. doi:10.1007/s00037-015-0114-7.

5 Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018. doi:10.1145/3170709.

6 Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Deterministic polynomial identity tests for multilinear bounded-read formulae. *Computational Complexity*, 24(4):695–776, 2015. doi:10.1007/s00037-015-0097-4.

7 Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000. doi:10.1145/337244.337257.

8 Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992. doi:10.1137/0221006.

9 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. doi:10.1145/62212.62241.

10 Vishwas Bhargava and Sumanta Ghosh. Improved hitting set for orbit of roabps. *Electron. Colloquium Comput. Complex.*, 28:62, 2021. URL: https://eccc.weizmann.ac.il/report/2021/062/.

11 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2144–2160. SIAM, 2020. doi:10.1137/1.9781611975994.132.

12 Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Information Processing Letters*, 8(5):234–235, 1979. doi:10.1016/0020-0190(79)90113-3.

13 Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory. https://pcwww.liv.ac.uk/~iken/teaching_sb/summer17/introtogct/gct.pdf, 2019.

14 Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5):32:1–32:29, 2018. doi:10.1145/3209663.

15 Daoud Bshouty and Nader H. Bshouty. On interpolating arithmetic read-once formulas with exponentiation. *J. Comput. Syst. Sci.*, 56(1):112–124, 1998. doi:10.1006/jcss.1997.1550.

16 Nader H. Bshouty, Thomas R. Hancock, and Lisa Hellerstein. Learning arithmetic read-once formulas. *SIAM J. Comput.*, 24(4):706–735, 1995. doi:10.1137/S009753979223664X.

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

18 Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

19 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs randomness for bounded depth arithmetic circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CCC.2018.13.

20 Rafael Mendes de Oliveira, Amir Shpilka, and Ben lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 25(2):455–505, 2016. doi:10.1007/s00037-016-0131-1.

21 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

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

23 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Commun. ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.

24 Michael A. Forbes. Some concrete questions on the border complexity of polynomials. https://www.youtube.com/watch?v=1HMogQIHT6Q, 2016.

25 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875. ACM, 2014. doi:10.1145/2591796.2591816.

26 Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1180–1192. ACM, 2018. doi:10.1145/3188745.3188792.

27 Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. Full version at http://arxiv.org/abs/1606.05050. doi:10.4230/LIPIcs.CCC.2016.32.

28 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018. doi:10.4086/toc.2018.v014a018.

29 Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. doi:10.1007/s00037-015-0103-x.

30 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. arXiv:1701.01717.

31 Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.APPROX/RANDOM.2020.4.

32 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness: Treading the borders. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 147–157. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00018.

33 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. doi:10.1137/140957123.

34 Ankit Gupta, Neeraj Kayal, and Satya Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 625–642, 2012.

35 Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *Computational Complexity*, 23(2):207–303, 2014. doi:10.1007/s00037-014-0085-0.

36 Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990. doi:10.1016/0196-6774(90)90014-6.

37 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 262–272. ACM, 1980. doi:10.1145/800141.804674.

38 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.

39 Kyriakos Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. doi:10.1137/0214050.

40 Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 280–291. IEEE, 2008.

41 Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 274–285. IEEE Computer Society, 2009. doi:10.1109/CCC.2009.18.

42 Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 643–662, 2012.

43 Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *Computational Complexity*, 28(4):749–828, 2019. doi:10.1007/s00037-019-00189-0.

44 Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):1–56, 2018.

45 Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 413–424. ACM, 2019. Full version at https://eccc.weizmann.ac.il/report/2018/191. doi:10.1145/3313276.3316360.

46 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.33.

47 Adam R. Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of Computing*, 2(10):185–206, 2006. doi:10.4086/toc.2006.v002a010.

48 Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001.

49 Mrinal Kumar. On the power of border of depth-3 arithmetic circuits. *ACM Trans. Comput. Theory*, 12(1):5:1–5:8, 2020. doi:10.1145/3371506.

50 Mrinal Kumar and Ramprasad Saptharishi. Hardness-Randomness tradeoffs for algebraic computation. *Bull. EATCS*, 129, 2019. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/591/599.

51 Joseph M. Landsberg. *Geometry and Complexity Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017. doi:10.1017/9781108183192.

52 Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theor. Comput. Sci.*, 66(1):1–14, 1989. doi:10.1016/0304-3975(89)90141-2.

53 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in vp$_e$ and $\sigma\pi\sigma$ circuits. *Electron. Colloquium Comput. Complex.*, 28:14, 2021. URL: https://eccc.weizmann.ac.il/report/2021/014.

54 Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Transactions on Computation Theory (TOCT)*, 10(3):1–11, 2018.

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

56 Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008. doi:10.1137/080718115.

57 Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. doi:10.4086/toc.2010.v006a007.

58 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. doi:10.1007/s00037-005-0188-8.

59 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. The power of depth 2 circuits over algebras. In Ravi Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 371–382. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2333.

60 Chandan Saha and Bhargav Thankey. Hitting sets for orbits of circuit classes and polynomial families. *Electron. Colloquium Comput. Complex.*, 28:15, 2021. URL: https://eccc.weizmann.ac.il/report/2021/015.

61 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github survey*, 2015. Available at https://github.com/dasarpmar/lowerbounds-survey.

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

63 Nitin Saxena. *Progress on Polynomial Identity Testing-II*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 131–146. Birkhäuser Basel, 2014. arXiv:1401.0976.

64 Nitin Saxena and C. Seshadhri. Blackbox Identity Testing for Bounded Top-Fanin Depth-3 Circuits: The Field Doesn't Matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012. doi:10.1137/10848232.

65 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM Journal on Computing*, 38(6):2130–2161, 2009.

66 Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. *Computational Complexity*, 24(3):477–532, 2015.

67 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. doi:10.1561/0400000039.

68 Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 31:1–31:53. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.31.

69 Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017. doi:10.1109/FOCS.2017.70.

70 Joseph Swernofsky. Tensor rank is hard to approximate. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 26:1–26:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.26.

71 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. doi:10.1137/0212043.