Polynomial Identity Testing for Read-4 Arithmetic Formulas

Nimrod Kaplan
Blavatnik School of Computer Science and AI, Tel Aviv University, Israel

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

Abstract

We present the first algorithms for polynomial identity testing (PIT) of read-4 arithmetic formulas in the non-multilinear setting. Specifically, we give a polynomial-time PIT algorithm in the whitebox model and a quasi-polynomial-time algorithm in the blackbox model. Since our techniques are based on proving *hardness of representation* results, we extend our algorithms to *orbits* of read-4 formulas under the action of the affine linear group. Prior to our work, no subexponential white- or blackbox algorithms were known for this class of formulas. All our results hold over any field $\mathbb{F}$ with $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq 5$.

Prior work addressed only restricted cases. Anderson, van Melkebeek, and Volkovich (*Computational Complexity*, 2015) studied *multilinear* read-k formulas, giving a polynomial-time whitebox PIT algorithm and a quasi-polynomial-time blackbox algorithm. Without the multilinearity restriction, Mahajan, Rao, and Sreenivasaiah (*TCS*, 2014) gave polynomial-time whitebox algorithms for read-2 and read-3 formulas, Prakriya (*Doctoral Thesis*, 2019) obtained quasi-polynomial-time blackbox PIT algorithm for both read-2 and read-3 formulas. Independently, Shamir (*Master's Thesis*, 2022) obtained a quasi-polynomial-time blackbox PIT algorithm for read-2 formulas. For *bounded-depth* read-k formulas, Agrawal, Saha, Saptharishi, and Saxena (*SICOMP*, 2016) obtained a polynomial-time blackbox algorithm in the non-multilinear case. The running time of their algorithm is $n^{k^{2^{\Delta}}}$ for read-k, depth-Δ formulas, and hence it is applicable only to constant depth.

Partial derivatives are a central tool in the study of deterministic PIT for bounded-read formulas. However, for non-multilinear RkF, differentiation may increase the number of reads. To address this, we develop new structural results that ensure "nice behavior" of derivatives. Specifically, we introduce a new *Fragmentation Lemma* that reduces the PIT problem for general RkFs to simpler models via differentiation. In addition, we define the notion of *dominating degree patterns* and show that, in certain cases, taking partial derivatives with respect to these patterns preserves the read count.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases algebraic complexity theory, polynomial identity testing, PIT, bounded read formulas

Digital Object Identifier 10.4230/LIPIcs.CCC.2026.25

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2026/076/ [27]

Funding This research was funded by the European Union (ERC, EACTP, 101142020). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

Acknowledgements We thank the anonymous referee who brought [39] to our attention.

COMPUTATIONAL COMPLEXITY CONFERENCE

1 Introduction

Arithmetic circuits are directed acyclic graphs that compute multivariate polynomials using the arithmetic operations $+$ and $\times$. Similarly, arithmetic formulas are circuits whose underlying graph is a tree. These are the most common models for studying the complexity of computing polynomials.

Polynomial Identity Testing (PIT) is a central problem in algebraic complexity theory that asks, given an arithmetic circuit or formula, whether it computes the identically zero polynomial. The problem is studied in two main settings. In the *whitebox* model, the circuit is explicitly given to the algorithm. In the *blackbox* model the algorithm can only make query access to the circuit. That is, it can only evaluate the polynomial at chosen inputs. Solving PIT in the blackbox model is equivalent to constructing a *hitting set* for the class of polynomials under consideration. A hitting set consists of a set of inputs such that any nonzero polynomial in the class evaluates to a nonzero value on at least one point in the set.

If randomness is allowed, the polynomial identity lemma (also known as the Ore-DeMillo-Lipton-Schwartz-Zippel lemma [38, 12, 45, 54]) guarantees a randomly chosen input from a sufficiently large grid will, with high probability, provide a nonzero evaluation point for the circuit. The main challenge in the field, and indeed, one of the most fundamental open problems in theoretical computer science, is to find a *deterministic* algorithm. PIT is the most general and well-studied algebraic derandomization problem, with wide range of applications, including primality testing [2, 3], parallel algorithms for perfect matchings [31, 15, 52], linear matroid intersection [23], derandomization of Noether normalization lemma [17, 37] and even the celebrated proof that IP equals PSPACE [32, 46].

Another reason for the importance of the PIT problem is its deep connection to circuit lower bounds. A series of works [24, 26, 1, 14, 11, 29, 20, 30, 7] established that derandomizing PIT is essentially equivalent to proving strong lower bounds for arithmetic circuits, in various settings. Furthermore, depth reduction results show that solving PIT even for highly restricted models such as depth-3 and depth-4 circuits [4, 28, 53, 21] would have major consequences.

Given its importance and the absence of deterministic algorithms in the general case, much research has focused on restricted models of computation. There is a rich body of work on PIT for small-depth circuits, read-once algebraic branching programs, bounded-read formulas, read-once determinants, and other restricted settings. For broader surveys on PIT and its connections, see [43, 51, 42, 44, 13].

One natural approach to derandomizing PIT for arithmetic formulas is to restrict the number of times each variable is read at the leaves. This line of work began with Shpilka and Volkovich [50], who provided a quasi-polynomial-size hitting set and a polynomial-time whitebox algorithm for read-once formulas, i.e., formulas in which each variable appears at most once. They also showed how to reconstruct (learn) such formulas deterministically [49]. Minahan and Volkovich completely settled the case of read-once formulas by providing a polynomial-sized hitting set. The works [49, 50] also introduced what is now known as the SV-generator, a k-wise independent polynomial map.

Anderson, van Melkebeek, and Volkovich significantly extended this line of work by studying *read-k multilinear formulas* (multilinear RkF for short),[1] for constant k. These are formulas in which each variable appears in at most k-leaves, and each gate computes

[1] We later refer to their model as *structurally-multilinear* to distinguish it from cases where multilinearity arises only through cancellations of nonmultilinear terms. In the literature this is often called *syntactically-multilinear*, but in our setting we wish to capture this "syntactic" behavior variable-wise and therefore use a different term.

a multilinear polynomial. They presented a polynomial-time whitebox algorithm and a quasi-polynomial-size hitting set. Notably, their result also implies a $2^{(1-\varepsilon)n}$-size hitting set for general multilinear formulas (without any restriction on the number of reads) of size $O(n)$.

As the results of [6] only hold for multilinear formulas, attempts were made to remove this restriction. Mahajan, Rao, and Sreenivasaiah [34] studied read-2 and read-3 formulas, and gave polynomial-time whitebox PIT algorithms for them. Prakriya [39] gave quasi-polynomial-size hitting sets for read-2 and read-3 formulas. Independently, Shamir [47] gave quasi-polynomial-size hitting sets for read-2 formulas.

In a different direction, Anderson et al. [5] studied read-k algebraic branching programs (Rk-ABPs, for short). An ABP is a model that subsumes formulas and is believed to be weaker than circuits but stronger than formulas. Any RkF can be represented as an Rk-ABP, meaning that each variable labels at most k edges (for more on ABPs see [51, 42]). In general though, in an ROABP each variable can appear on multiple edges, as long as they all appear in the *same layer*. Thus, it is a much stronger model than read-once formulas. Anderson et al. gave a subexponential-time whitebox[2] PIT algorithm for Rk-ABPs. Specifically, their algorithm runs in time $2^{\tilde{O}(n^{1-1/2^{k-1}})}$, and yields an algorithm with similar running time for RkFs.

Other than these results, the problem of derandomizing PIT for RkFs remains widely open, both in the whitebox and blackbox settings.

In this work, we design a deterministic quasi-polynomial-time blackbox PIT algorithm for R4Fs. Before our result, no nontrivial deterministic blackbox or whitebox PIT algorithms were known for R4Fs, apart from the slightly subexponential Rk-ABP algorithm of [5].

1.1 Our Results

All our results are stated in terms of *independent polynomial maps*, a key tool that allows us to reduce the number of variables while preserving nonzeroness. We denote with $\mathcal{G}_m$ a polynomial map with seed length m.[3]

The main technical contribution of this work is a quasi-polynomial-time blackbox algorithm for R4Fs. The class R4F is of particular interest, as four is the minimum number of reads required for a non-structural variable to appear.

▶ **Definition 1** (Structural variable, Structural formula)**.** *For some algebraic formula F, a variable $x_i \in var(F)$ is said to be structural if for each pair of gates $v, u \in F$, such that v is an ancestor of u, it holds that $\deg_{x_i}(v) \geq \deg_{x_i}(u)$.*

A formula F is structural if every $x_i \in var(F)$ is structural. ⌟

Non-structural variables may cause "massive cancellation" which is the core difficulty in verifying polynomial identities.

For a formula F over the variables $\mathbf{x}$ we denote with $p_F(\mathbf{x})$ the polynomial computed at the root of the formula.

All our results hold over any field $\mathbb{F}$ such that $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) \geq 5$. This restriction arises because our argument uses partial derivatives of order at most four, which may vanish identically over fields of smaller characteristic. Henceforth, we omit explicit mention of the underlying field. Note, however, that constructing a hitting set may require passing to an extension field if $|\mathbb{F}|$ is too small.

[2] The algorithm uses its whitebox access only to determine the order in which the variables are read. Such a model is sometimes referred to in the PIT literature as *grey-box*.

[3] Explicit constructions such as the SV-generator [50] and the RFE-generator [25] are known.

▶ **Theorem 2** (Main blackbox result). *Let $p_F(\mathbf{x})$ be an n-variate polynomial computable by an* R4F *F, over a field $\mathbb{F}$. There exists an absolute constant c_2, such that $p_F(\mathbf{x}) \not\equiv 0$ if and only if*

$$p_F \circ \mathcal{G}_{9\log n + c_2} \not\equiv 0\,.$$

We also present a polynomial-time whitebox algorithm for the same model.

▶ **Theorem 3** (Main whitebox result). *Let F be an n-variate* R4F *formula over a field $\mathbb{F}$. There exists an algorithm that, given whitebox access to F, checks in time poly(n) whether $p_F \equiv 0$.*

For R3Fs, we obtain a constant-factor improvement over the seed length used for the hitting set of [39] (and also compared to Theorem 2)

▶ **Theorem 4.** *Let $F \in$* R3F *be nonzero. Then, there exists a constant c_4*

$$p_F \circ \mathcal{G}_{2\log n + c_4} \not\equiv 0\,.$$

We next give a constant-factor improvement in the seed length over the result of [47], and an additive constant improvement in the seed length over the result of [39] (and also compared to Theorem 4).

▶ **Theorem 5.** *Let $F \in$* R2F *be nonzero. Then*

$$p_F \circ \mathcal{G}_{\log n + 4} \not\equiv 0\,.$$

Having established deterministic hitting sets for bounded-read formulas, we next extend these results to orbit classes. I.e., we compose polynomials with *full-rank affine transformations*. The key observation is that hardness of representation results for the base class carry over to the corresponding orbit class. This allows us to "lift" results from the backbone class (the class whose being composed with the affine transformation) to the orbit class. In all results below, $N \le n$ denotes the number of variables of the backbone formulas.

We begin by generalizing the results for sums of ROFs from [50] and [9].

▶ **Theorem 6.** *Let $F_1(\mathbf{y}), F_2(\mathbf{y}), \ldots, F_k(\mathbf{y}) \in$* ROF*, and let $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$. Then*

$$\sum_{i=1}^{k} p_{F_i}(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad \sum_{i=1}^{k} p_{F_i}(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{\log N + 3k} \not\equiv 0\,.$$

If $k = 2$ then we can compose with $\mathcal{G}_{\log N + 3}$ instead of $\mathcal{G}_{\log N + 6}$.

We next prove the result for orbits of structurally multilinear polynomials. Our proof closely follows the argument of [6], except that we use our version of the Fragmentation Lemma rather than the version introduced in that work.

▶ **Theorem 7.** *Let $F(\mathbf{y}) \in$* RkF *be a structurally multilinear formula, and let $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$. Then, for $r_k \coloneqq r_{2,2k} = k^{o(k)}$,*

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{r_k + k\log N} \not\equiv 0\,.$$

Finally, we lift our results for general read-2, read-3, and read-4 formulas to their corresponding orbit classes.

▶ **Theorem 8.** *Let* $F(\mathbf{y}) \in \mathsf{R2F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then*

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{3\log N+5} \not\equiv 0.$$

▶ **Theorem 9.** *Let* $F(\mathbf{y}) \in \mathsf{R3F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then, there exists a constant* c_9

$$p_F(\mathbf{y}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{6\log N+c_9} \not\equiv 0\,.$$

▶ **Theorem 10.** *Let* $F(\mathbf{y}) \in \mathsf{R4F}$ *and let* $(\boldsymbol{A}, \boldsymbol{\beta}) \in GL_n^{aff}(\mathbb{F})$*. Then, there exists a constant* c_{10}

$$p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \not\equiv 0 \quad \Rightarrow \quad p_F(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{\beta}) \circ \mathcal{G}_{12\log N+c_{10}} \not\equiv 0\,.$$

To the best of our knowledge, prior to this work, no nontrivial blackbox PIT algorithm was known for any of these classes.

An important tool in our proofs, and indeed in most previous work on PIT for bounded-read formulas, is the use of partial derivatives and the analysis of the resulting simplified formula. While this approach is effective for *multilinear* formulas, this is no longer the case for formulas with non-multilinear structure; in fact, differentiation may result in a more complex formula than the original one.

In this work, we introduce several structural tools that enable the identification of partial derivatives that do achieve the desired simplification. First, we define two key sets of gates: the **R(k − 1)F-frontier** , which identifies "maximal" R(k − 1)F subformulas, and the set $\boldsymbol{\mathcal{F}^{(2)}}$, which captures "minimal" RkF subformulas.

▶ **Definition 11** (R(k − 1)F Frontier)**.** *Let* F *be an* RkF*. The* R(k − 1)F *Frontier of* F*, denoted* $\mathcal{F}_F$*, is the set of the topmost gates that compute an* $R(k-1)P$ *polynomial. That is, there is no other gate in the path from these gates to the root that computes an* $R(k-1)P$ *polynomial. Whenever* F *is clear from context, we write* $\mathcal{F}$ *instead of* $\mathcal{F}_F$*.* ⌟

▶ **Definition 12** ($\mathcal{F}^{(2)}$)**.** *Let* F *in* RkF*. Denote by* $\mathcal{F}_F^{(2)}$ *the set of all the gates in* F *which have two children in* $\mathcal{F}_F$*. Whenever* F *is clear from context, we write* $\mathcal{F}^{(2)}$ *instead of* $\mathcal{F}_F^{(2)}$*.* ⌟

We denote, for $d \in \mathbb{N}$

$$\partial_{x_i^d}(p) \coloneqq \frac{\partial^d p}{\partial x_i^d} \quad \text{and} \quad \partial_{x_i^{\deg}}(p) \coloneqq \partial_{x_i^{\deg_{x_i}(p)}}(p)$$

to mean the derivative of p taken with respect to x_i as many times as the degree of x_i in p.

Additionally, we define the notion of **stable variables**:

▶ **Definition 13** (Stable set of a gate in an RkF, $U^{(s)}(v)$)**.** *Let* $F \in \mathsf{RkF}$ *and let* $v \in F$*. We denote by* $Unv_F(v)$ *the set of unvisited children of multiplication gates along the path from* v *to the root of* F*. The* stable set *of* v*, denoted* $U_F^{(s)}(v) \subseteq \mathbf{x}$*, is the set of variables* $x_i \in var(v)$ *such that*

$$\partial_{x_i^{\deg}} p_F = \partial_{x_i^{\deg}} p_v \cdot f \quad \text{or} \quad \partial_{x_i^{\deg}} p_F = f\,,$$

where $f = \prod_{i=1}^{m} f_i$ *is an RkP and the* f_i*s are computed by distinct subformulas of gates in* $Unv_F(v)$*. When* F *is clear from context, we write* $U^{(s)}(v)$*.* ⌟

Differentiating with respect to a stable variable of a gate guarantees that the resulting structure above that gate is simplified.

These tools are combined and applied in our **Fragmentation Lemma**, which provides a method for simplifying the structure of a general RkF via differentiation. This lemma generalizes previous variants from earlier work and, in comparison, requires fewer partial derivatives to achieve the desired simplification.

Moreover, we introduce the concept of **dominating degree patterns**.

▶ **Definition 14** (Dominating Degree Patterns, Informal). *A degree pattern is a partial exponent vector $\mathfrak{e} \in \mathbb{N}^J$ defined over a domain $J \subseteq [n]$. We say $\mathfrak{e}$ is* dominating *in a polynomial f if f contains a monomial whose exponent vector coincides with $\mathfrak{e}$ on J, while any monomial in f with a strictly higher degree in one variable of J must necessarily have a strictly lower degree in another.*

We then demonstrate that taking partial derivatives with respect to such dominating degree patterns preserves the read-count of all other variables, provided each variable in the pattern is read at most twice and that $|J| \leq 3$.

▶ **Lemma 15.** *Let $S \subseteq [n]$ be of size $|S| \leq 3$. Let $F \in$ RkF be such that for every $i \in S$, $Read_F(x_i) \leq 2$. Let $\mathfrak{e} \in \{1,2\}^S$ be a dominating degree pattern of F with domain S. Then, $\partial_{\mathfrak{e}} F \in$ RkF, and moreover it is composed entirely of disjoint subformulas of F.*

1.2 Comparison to previous work

There are several works concerning bounded-read formulas with more than one read per variable.

Shpilka and Volkovich [50] studied sums of k ROFs. They gave an $n^{O(k)}$ whitebox algorithm and a hitting set of size $n^{O(\log n+k)}$. Their result is actually more flexible and allows to substitute univariable polynomials $f_i(x_i)$ for the x_i-s. This work also introduced the SV-generator and the hardness of representation technique (HoR for short). In a beautiful work, Minahan and Volkovich proved that a 1-independent map hits ROFs. Using that they obtained an $n^{O(k)}$ blackbox PIT algorithm for sum of k ROFs.

We note that a sum of k ROFs is naturally a *multilinear read-k formula*, but of a very restricted structure. Anderson, van Melkebeek and Volkovich presented a polynomial-time whitebox algorithm and a quasi-polynomial-size hitting set for the general model of multilinear read-k formulas. We later explain what the difficulty is in extending their techniques to the non-multilinear case.

Going beyond multilinearity, Mahajan, Rao, and Sreenivasaiah [34] studied read-2 and read-3 formulas, without multilinearity restrictions, and gave polynomial-time whitebox PIT algorithms for them. In his Ph.D. thesis, Prakriya [39] gave quasi-polynomial-size hitting sets for read-2 and read-3 formulas. A central component of his approach is a reduction from blackbox PIT for general RkFs to blackbox PIT for $\sum^4 \mathsf{R(k-1)F}$. This tool serves as an alternative to our fragmentation lemma. The main difference is the target model, while Prakriya's reduction targets $\sum^4 \mathsf{R(k-1)F}$, our lemma reduces to certain derivatives of $\sum^2 \mathsf{R(k-1)F}$. Furthermore, Prakriya's also introduced a generalization to the "shattering" lemma from [6]. Independently, Shamir, in his M.Sc. thesis [47], obtained quasi-polynomial-size hitting sets for read-2 formulas.

A different multi-read restricted model was considered by Mahajan, Rao and Sreenivasaiah [33]. Specifically, they considered what we term as $\sum^2 \prod \mathsf{ROF}$. That is, a sum of two terms, each is a product of ROFs. For this model they gave a whitebox PIT algorithm.

We note that while this model can allow unbounded number of reads, it is a much easier model, from the PIT perspective. Indeed, in the whitebox model it is quite simple to find irreducible factors of ROFs and then all we have to do is normalize each factor and compare multiplicities and evaluation at a nonzero point common to all the ROFs.

Bisht, Gupta and Volkovich [9] gave a polynomial-time blackbox PIT algorithm for $\sum^2 \prod \mathsf{ROF}$. In the same work, they also developed polynomial-time and quasi-polynomial-time blackbox algorithms for the classes of sums of three powers of ROFs and sums of three powers of multilinear RkFs, respectively. Thus, these models allow more than 2 reads, and they don't compute multilinear polynomials, yet there is a very strong restriction on their structure.

To summarize, prior work either dealt with multilinear bounded-read formulas, or bounded-read formulas with a very restricted structure. In particular, no whitebox or blackbox PIT was known for sums of two R2Fs, which is a very restricted form of R4Fs.

Algebraic branching programs (ABPs) are layered graphs with a source and sink nodes, in which variables label edges that go between layers and the polynomial computed by the ABP is the sum over all source-sink paths of the product of the labels of the edges in the path. ABPs form an important model of computation that is (believed to be) stronger than formulas and weaker than circuits. In this model, a read-once ABPs is an ABP in which each variable can appear on edges between two specific adjacent layers. This model is considerably stronger than ROFs as an ROF can be computed by an ABP in which a variable labels only a single edge.

Raz and Shpilka [40] provided a polynomial time whitebox PIT algorithm for ROABPs. This immediately implies such an algorithm for read-once formulas (ROFs), although in the case of formulas the problem is much easier. In the blackbox setting, Forbes and Shpilka [16] gave the first quasi-polynomial time blackbox algorithm for ROABPs. This started a long line of work on read-once models of ABPs – the usual ROABP model, commutative-ROABP, any-order ROABPs, bounded-width ROABPs, sums of ROABPs etc. See [13] for a recent survey on PIT that contains many of the state of the art results on ROABPs.

We note that according to the basic definition, ROABPs compute multilinear polynomials. However, similar to [50], one can allow univariate polynomials to label edges instead of just variables. This results in non-multilinear computations, that are still as structured as ROABPs.

ABPs that read each variable more than once (i.e., in more than one layer), were first considered by [22] who studied PIT for *sum* of ROABPs. They gave a a whitebox PIT algorithm for the sum of k ROABPs that runs in time $w^{O(2^k)} \cdot (nd)^{O(k)}$, where d is an upper bound on the univariates degree, k is the number of summands, and w is the maximal width (i.e. size of layer) of any of the ROABPs in the sum. In the blackbox setting they gave a hitting set of size $(ndw)^{k2^k \log(ndw)}$ for the model. This model allows reading variables in different layers, and allows non-multilinearity by enabling univariates on edges, yet its underlying structure still comes form that of a multilinear sum of ROABPs. In that respect, this model can be seen as the ABP analog of the sum-of-ROFs results of [50, 36] .

Other restricted models allowing multiple reads considered bounded depth formulas. There is a vast literature on PIT for bounded depth models, but as we are concerned with formulas without any depth restriction we refer the readers to the surveys [43, 51, 44, 13] as well as to Forbes' thesis [19].

Finally, the aforementioned work of [5] considered unrestricted read-k ABPs, which is a considerably stronger model than read-k formulas. They constructed a gray-box PIT algorithm running in time $2^{\tilde{O}(n^{1-1/2^{k-1}})}$ for this model. This is a stronger model than what

we consider, but we achieve a polynomial time whitebox algorithm (compared to theirs slightly sub-exponential algorithm) and a quasi-polynomial-size hitting set, whereas their algorithm needs to know the order of the reads so it does not work in a blackbox model, and of course, it is also slightly sub-exponential, compared to our quasi-polynomial-size hitting set.

To conclude, besides the slightly sub-exponential PIT algorithm of [5], PIT algorithms for multi-read ABPs dealt with substitutions of univariates to models based on multilinear polynomials.

Another line of research, motivated by the fact that simple polynomials, when composed with full-rank affine transformations, form dense subsets in broader and more interesting classes. This line of work started with the work of Bringmann, Ikenmeyer and Zuiddam [10] who proved that orbits of width-2 ROABPs are dense in the class of poly-size formulas. Subsequent work, considered orbits of classes such as ROFs [35], sparse polynomials [35, 41] and ROABPs [41, 8]. Of particular relevance is the work by Medini and Shpilka [35] who proved that orbits of ROFs are dense within poly-size arithmetic formulas. They also gave hitting sets and reconstruction algorithms for polynomials in such orbits. As Forbes and Shpilka [18] noted, if these hitting sets could be made *robust*[4] then they would imply hitting set for *general arithmetic formulas*. Thus, exploring hitting sets for such simple classes is an extremely interesting question, with the goal being developing that would eventually lead to robust hitting sets. We note that in the aforementioned work [6], the authors allowed variables to be replaced with arbitrary sparse polynomial (of polynomial degree), but required that the children of every multiplication node in the substituted multilinear read-k formula are variable disjoint. Thus, this allows more flexibility in the sense that more general polynomials than linear functions are being substituted, however there is the strong restriction of remaining multilinear. In contrast, when we compose, even multilinear read-k formulas with a linear transformation, the result is unlikely to remain multilinear.

2 Proof overview and techniques

As mentioned earlier, the main difficulty in analyzing read-4 formulas is that variables cease to behave *structurally* (recall Definition 1). For read-3 or fewer read formulas, every variable occurs in a "consistent" way: if a node v depends on a variable x and u is a descendant of v, then the degree of x in the subformula at u cannot exceed its degree at v. For read-4 formulas this is no longer the case, and this loss of structural behavior is the source of most of the technical difficulties we face.

One reason this loss of structure is problematic is that previous algorithms crucially relied on taking partial derivatives of the formula. In the multilinear setting, differentiating with respect to a variable only simplifies the formula, while not increasing the number of reads. This is captured by the *fragmentation lemma* of [6]. However, once variables can appear with higher degrees, differentiation may destroy the formula's bounded-read property. For example, if a subformula contains terms such as $(Ax + B)(Cx + D)$ and the quadratic term ACx^2 cancels out, then the formula depends on x with degree 2 at some gates and degree 1 at some of their parents. Differentiating twice with respect to x yields zero, while differentiating once produces a term like $AD + BC$, which may involve many more reads than the original subformula. Thus, the derivative can increase the read count beyond 4, breaking the inductive structure that underlies previous analyses.

[4] We say that a hitting set $\mathcal{H}$ is *robust* for a class $\mathcal{C}$ if there exists a constant $c > 0$ and a fixed norm $\|\cdot\|$ on $\mathbb{C}[\boldsymbol{x}]$ such that for every $f \in \mathcal{C}$ there exists $\boldsymbol{\alpha} \in \mathcal{H}$ satisfying $|f(\boldsymbol{\alpha})| \geq c \cdot \|f\|$.

It is not surprising that cancellations lie at the heart of the difficulty, since they are precisely what make PIT challenging in general. Many efficient computations for example, the Determinant polynomial, rely on intricate patterns of cancellation that obscure the underlying algebraic structure.

At a high level, and omitting many technical details, our approach is to isolate and control these cancellations. We first show that if a formula has only few structural variables, then we can reduce to the extreme case in which *no* variable is structural. Furthermore, we can assume that all cancellations occur at the top gate. This reduction yields formulas of the form $\sum^2 \mathsf{R2F}$, where the resulting polynomial is multilinear even though each R2F has degree 2 in every variable. We refer to such formulas as *totally non-structural*. Analyzing the patterns of cancellation that arise in this setting constitutes the main technical component of our work.

We shall now expand more on the ideas that we use to handle these obstacles. Due to space constraints, we provide only a very high-level overview here; full details appear in the full version [27].

The first tool that we need is a *fragmentation lemma*. Fragmentation lemmas, as termed in [6], aim to reduce the problem of hitting the class RkF to that of hitting the class $\sum^2 \mathsf{R(k-1)F}$. Such lemmas were instrumental to the works of [6] and [47]. In our case, by concentrating on $\mathcal{F}^{(2)}$ (recall Definition 12), i.e., gates whose children are read-3 but they themselves are read-4, we prove that by taking an appropriate partial derivative with respect to a stable variable of one of these first read-4 gates, simplifies the formula, thus achieving *fragmentation*.

▶ **Lemma 16** (Fragmentation Lemma). *Let F be a nonzero* RkF *and set $t \coloneqq |\mathcal{F}_F^{(2)}|$. Write $\mathcal{F}_F^{(2)} = \{v_j\}_{j=1}^t$, and let $I = \{i_j\}_{j=1}^t$ with $x_{i_j} \in U^{(s)}(v_j)$. Denote*

$$\mathcal{D}_I \coloneqq \left\{ \partial_{x_{i_j}^{\deg}} p_{v_j} : j = 1, \ldots, t \right\}.$$

If $\mathcal{H}$ is a hitting set for $\mathcal{D}_I \cup \mathsf{R(k-1)F}$, then[5]

$$p_F \circ (\mathcal{H} + \mathcal{G}_{\log t+1}) \not\equiv 0.$$

Applying this procedure to R2Fs reduces the formula to a $\sum^2$ ROFs, which is hit by $\mathcal{G}_3$, which is how we obtain Theorem 5. In a similar vain, to prove Theorem 4, we again use our fragmentation lemma to reduce the PIT problem for R3Fs to PIT for an R3Fs of the form $\sum^2 \mathsf{R2F}$. If the sum is multilinear, then by the following claim it is structurally multilinear, and the result of [6] applies directly.

▷ Claim 17. Let $F \in \mathsf{R3F}$. Then, there exists a structural $F' \in \mathsf{RkF}$ such that $p_F = p_{F'}$ and for every $x_i \in \boldsymbol{x} : \mathrm{Read}_{x_i}(F') \leq \mathrm{Read}_{x_i}(F)$.

Otherwise, some variable has degree 2, implying that its degrees in the two summands are unbalanced. In this case, taking the second-order partial derivative with respect to that variable reduces the problem to hitting a nonzero R2F.

2.1 Proof Overview of the Blackbox Algorithm (Theorem 2)

The following sketch (Figure 1) outlines the structure of the proof and the dependencies among its key components.

[5] Whenever we write a sum of generators, we mean they are defined over disjoint sets of variables.

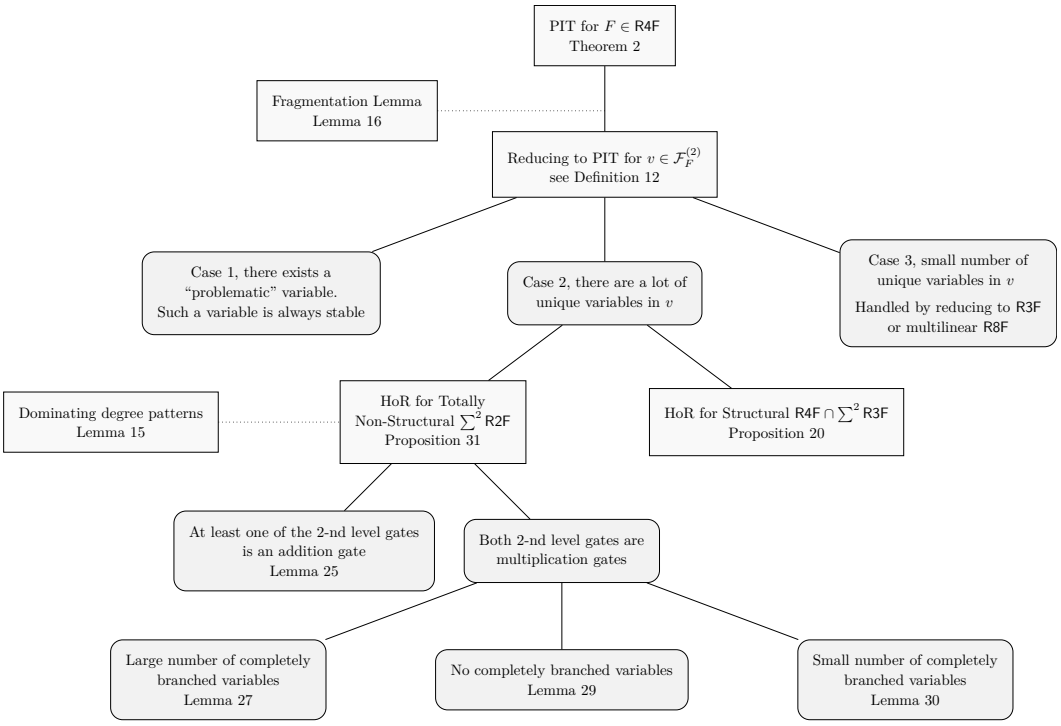

Figure 1 Structure of the proof of Theorem 2.

The fragmentation procedure discussed above allows us to focus our attention on a gate that is read-4 but that both its children are read-3.

If v is a multiplication gate then a hitting set for read-3 formulas hits p_v. When v is an addition gate we use hardness of representation together with restrictions by a generic assignment to obtain a hitting set. In the following part we give a short description of these techniques and how they are used together.

2.1.1 Hardness of Representation, Restrictions and Preserving Assignments

An idea that considerably simplifies the analysis is the use of *generic assignments* to a subset of the variables. A generic assignment is an assignment that does not alter any algebraic property[6] of the polynomials under consideration.

In the *hardness of representation* technique, we need to prove that a formula of a certain structure, cannot compute any polynomial divisible by the monomial $\mathcal{P}_n = \prod_{i=1}^{n} x_i$. Once such a result is proved, there is a way to use it to construct a hitting set. This technique has been successfully employed in several prior works on bounded read formulas [50, 6, 9, 39].

In general, one may observe that the monomial $\mathcal{P}_n$ can be computed even by a very restricted class of polynomials, such as the class of ROFs. Thus, hardness of representation for a class of polynomials $\mathcal{C}$ is obtained by identifying a collection of algebraic properties associated with polynomials in the class. More precisely, for every $p \in \mathcal{C}$, these properties are captured by a finite set of polynomials related to p which are simpler than p, for example, subformulas of F_p. The goal is to ensure that if $p \in \mathcal{C}$ is translated by a common nonzero of this finite set of polynomials, then the resulting polynomial is not divisible by $\mathcal{P}_n$.

[6] An algebraic property is one that can be expressed by a finite set of polynomial equalities.

For example, [48] showed that in order to prove hardness of representation for sums of ROFs, it suffices to translate by a common nonzero $\boldsymbol{\sigma}$ of all the partial derivatives of the individual ROFs. This condition is captured by the algebraic property of being **0**-justified, which ensures that no variable on which some term in the sum depends disappears after another variable is set to 0. The translation $\boldsymbol{\sigma}$ is then called a *justifying assignment*.

We call assignments that preserve a desired property of the polynomial under consideration *preserving assignments*. In this work, we use two additional types of preserving assignments. The first type, introduced in [9], is called an *irreducibility-preserving* assignment. A polynomial is **0**-irreducible if it is irreducible and remains irreducible after setting any strict subset of its variables to 0.

In this work, we introduce the notion of *degree-preserving* assignments. Such assignments preserve the degrees of the remaining variables in the polynomial for which they are constructed. Thus, they may be viewed as a stronger form of justifying assignments. A polynomial translated by such an assignment has its individual degrees preserved under substitutions of variables by 0, and we call such a polynomial **0**-preserved.

In the following subsections we define the sets of polynomials that are used to describe these preservation properties. In order to prove hardness of representation for $\mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ we will first translate the polynomial by a common nonzero of certain polynomials from these sets.

In our analysis we combine this approach with generic assignments. Given a generic assignment we can *restrict* a formula to a subset of its variables, by substituting values to the other variables according to the assignment and simplifying the resulting formula. Observe that if the monomial $\mathcal{P}_n$ divides the original polynomial, then $\mathcal{P}_S = \prod_{i \in S} x_i$ must divide the restricted one. Hence, it is enough to prove hardness of representation results for the restricted polynomial.

The main advantage of restrictions is that it greatly simplifies the formula and enables us to reduce the analysis to arguing about polynomials in few variables.[7]

For example, to restrict our attention to p_v we may restrict the variables not appearing in its subtree.

We stress that the generic assignment is only used for the analysis, and not for the construction of the hitting set.

The difficult case, as explained, is when there are many non-structural variables, but let us first explain what we do in the mostly structural case.

2.1.2 The structural case

In the structural case, we use two sets of polynomials in order to find a translation that yields hardness of representation. The first set was introduced in [6] and is used to prove hardness of representation for structurally multilinear RkFs. Throughout, when we write $w \in F$ for a formula F, we view F as the set of its gates.

▶ **Definition 18** ($\mathcal{Q}_{F_1,F_2,\ldots,F_m}$)**.** *Let $F_1, F_2, \ldots, F_m$, be algebraic formulas such that F_i is an $\mathsf{Rk_iF}$ and let $k = \sum_{i\in[m]} k_i$. Define*

$$b_{m,k} := (k - m + 1) \cdot 4k \cdot (k + 1)^2 .$$

[7] For us, "few" can be 5^8 many variables, but in most cases we will eventually reduce to polynomials on at most 4 variables.

Then we define

$$\mathcal{Q}_{F_1,F_2,\ldots,F_m} \coloneqq \left\{ \partial_{Y^{\deg}}(p_w)\big|_{Z\leftarrow \mathbf{0}} \;\middle|\; w \in \bigcup_{i\in[m]} F_i,\ Z\subseteq[n],\ Y\subseteq[n]\setminus Z,\ |Z\cup Y|\le b_{m,k} \right\}.$$

In words, F_w is a subformula of some F_i, and $Z, Y \subseteq [n]$ are disjoint subsets satisfying $|Z \cup Y| \le b_{m,k}$. ⌟

The second set is defined as follows.

▶ **Definition 19.** *Let F, G be algebraic formulas. Define*

$$\mathcal{T} \coloneqq \left\{ \partial_{x_i^{\deg}}\left(\partial_{x_j^{\deg}} p\right) \;\middle|\; p \in \{p_F, p_G\},\ x_i, x_j \in var(p) \right\}.^{8}$$

If we translate $p \in \mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ by a common nonzero of the two sets above, then hardness of representation is easy to prove in each of the following two cases:

1. "Many" variables appear with degree 1 in p.
2. The degrees in the two summands are unbalanced.

Thus, by removing the "few" variables with degree 1 in p using a generic assignment, we are left with the case where $p \in \sum^2 \mathsf{R2F}$ and every variable has individual degree 2 in both summands. To handle this case, we introduce the notion of *splitness* and show that it implies hardness of representation.

Splitting a polynomial $h = f + g$ with respect to variables x_i and x_j corresponds to transforming h into a polynomial of the form $\widetilde{h} = f_i f_j + g_i g_j$, where f_i and g_i do not depend on x_j, and f_j and g_j do not depend on x_i. We achieve this by taking a partial derivative with respect to a carefully chosen variable x_t.

More precisely, the three variables i, j, t depend on each other, and on the formula computing h, and we find them using combinatorial analysis of labeled trees, relying on the well-known Erdős–Szekeres theorem.

We then prove the main point which is that a split polynomial cannot be divisible by a high degree monomial. Going back to h we obtain the same result and hence conclude hardness of representation in the structural case.

▶ **Proposition 20.** *Let F, G be $\mathsf{R3F}$s such that the formula $F + G$ is a structural $\mathsf{R4F}$. Let U be a set containing all the read-4 variables in the formula $F + G$.*

Assume $U \neq \emptyset$ and that no nonzero polynomial in $\mathcal{Q}_{F,G} \cup \mathcal{T}$ vanishes at $\mathbf{0}$.

Then, if $n \ge r_{2,6} + 5^8$, there exists $t \in U$ such that $\mathcal{P}_{[n]\setminus\{t\}} \nmid \partial_{x_t^{\deg}}(F + G)$.

Let us now switch gears and consider the non-structural case.

2.1.3 Totally non-structural case

First, observe that a totally non-structural polynomial in $\mathsf{R4F} \cap \sum^2 \mathsf{R3F}$ must have the structure of a $\sum^2 \mathsf{R2F}$. Indeed, each of the two summands has degree 2 in every variable, yet all such degree-2 monomials cancel. In particular, the resulting polynomial is multilinear. In this part, we use this special structure to prove hardness of representation.

---

[8] Observe that $\partial_{x_i^{\deg}}\left(\partial_{x_j^{\deg}} p\right) \neq \partial_{x_i^{\deg} x_j^{\deg}} p$. On the left-hand side, we differentiate with respect to x_i as many times as its degree in $\partial_{x_j^{\deg}} p$, whereas on the right-hand side, we differentiate as many times as its degree in p.

We now describe the sets of polynomials that we use in the totally non-structural case to obtain hardness of representation. We denote the root of F by o, its left child by $(o)_{\mathrm{L}}$, and its right child by $(o)_{\mathrm{R}}$. We also use the following notation, recalling Definition 14:

$$\partial_{I^{\mathrm{dom}}} p_F = \left\{ \partial_{\mathfrak{c}} p_F \;\middle|\; \begin{array}{c} \mathfrak{c} \text{ is a dominating degree} \\ \text{pattern in } p_F \text{ with domain } I \end{array} \right\} .$$

▶ **Definition 21** ($\mathcal{A}_F$)**.** *Let F be an algebraic formula and $t \in \mathbb{N}$. Define,*

$$\mathcal{A}_F^t \coloneqq \bigcup_{I \in \binom{[n]}{\leq t}} \partial_{I^{dom}} p_F \ ,$$

In words, this is the set obtained by taking all possible derivatives according to all dominating degree patterns of size at most t. ⌟

▶ **Definition 22** ($\mathcal{E}_F^m$)**.** *Let F be an algebraic formula and let $m \in \mathbb{N}$. Define:*

$$\mathcal{E}_F^m \coloneqq \bigcup_{u \in F_{(o)_L} \cup F_{(o)_R}} \mathcal{A}_{F_u}^m .$$ ⌟

The utility of these definitions is captured in the next claim.

▷ Claim 23. Let F be an algebraic formula. If no polynomial in $\mathcal{E}_F^{m+1}$ vanishes at $\mathbf{0}$, then, for every subformula of $F_{(o)_{\mathrm{L}}}$ or $F_{(o)_{\mathrm{R}}}$, every derivative corresponding to a dominating degree pattern whose domain has size at most m is $\mathbf{0}$-preserved.

Additionally, the next set is used to make the ROF subformulas of the polynomial under consideration $\mathbf{0}$-irreducible.

▶ **Definition 24** ($\mathcal{C}_F$)**.** *Let F be an algebraic formula. For any pair of gates $u, v \in F$ such that u is an ancestor of v, let $\gamma_v \in \mathbb{F}$ denote the additive constant of the gate v. Define*

$$\mathcal{C}_u \coloneqq \left\{ p_{F_u|_{v=\gamma_v}} \;\middle|\; v \in F_u \right\},$$

In words, $\mathcal{C}_u$ is obtained by replacing, for each descendant v of u, the subtree rooted at v with a leaf labeled by the scalar γ_v and then reducing the formula.

We further define

$$\mathcal{C}_F \coloneqq \bigcup_{u \in F_{(o)_L} \cup F_{(o)_R}} \mathcal{C}_u \, .$$

When the formula F is clear from context, we simply write $\mathcal{C}$. ⌟

Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Since each child of the root of F computes a polynomial of degree 2 in every variable, the first common gate of the two leaves labeled by any variable x_i in $F_{(o)_{\mathrm{L}}}$ must be a multiplication gate, and the same holds in $F_{(o)_{\mathrm{R}}}$. Consequently, if, say, $(o)_{\mathrm{L}}$ is an addition gate, then there must exist variables $x_i, x_j \in \boldsymbol{x}$ whose first common gate is $(o)_{\mathrm{L}}$. Taking a partial derivative with respect to these two variables then reduces the problem to PIT for R2Fs.

▶ **Lemma 25.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that $(o)_L$ is an addition gate. Moreover, suppose that no nonzero polynomial in $\mathcal{A}^2_{F_{(o)_L}} \cup \mathcal{A}^2_{F_{(o)_R}}$ vanishes at $\mathbf{0}$. Then, if $n \geq 3$, we have $\mathcal{P}_n \nmid p_F$.*

Hence, the challenging case occurs when both $(o)_{\mathrm{L}}$ and $(o)_{\mathrm{R}}$ are multiplication gates. To handle this case, we introduce the notion of *branched variables*. A variable x_i *branches* in a product gate u if each child of u computes a linear polynomial in x_i. A variable that branches in the two subformulas computed by the children of the root is called *completely branched*, while one that branches in only one child is *partially branched*.

Note that if x_1 completely branches in F then F computes a polynomial of the form $(a_1x_1 + b_1)(a_2x_1 + b_2) + (a_3x_1 + b_3)(a_4x_1 + b_4)$, where the a_is and b_js are subformulas of F.

We further divide the analysis into three cases:

1. there exists a "large" set of variables that branch completely in F;
2. every variable that branches in F branches only partially;
3. the set of completely branched variables is "small".

To handle the first case, we first strengthen, in an appropriate sense, [9, Theorem 1]. This theorem provides a PIT algorithm for the class $\sum^2 \prod \mathsf{ROF}$. However, the proof does not proceed via a hardness of representation argument, which is what we require, since we already used a generic assignment in order to reduce to the $\sum^2 \mathsf{R2F}$ case. Accordingly, we first establish such a hardness of representation result.

▶ **Lemma 26.** *Let $F_1, F_2 \in \prod^k \mathsf{ROF}$ such that $F_i = \prod_{j=1}^{m_i} f_{i,j}$, where all $f_{i,j}$ (for $i \in [2]$, $j \in [m_i]$) are irreducible. Moreover, assume these irreducible factors are $\mathbf{0}$-irreducible and $\mathbf{0}$-preserved. Then, if $n \geq 2k + 1$, we have $\mathcal{P}_n \nmid p_{F_1} + p_{F_1}$.*

Now assume there exists a set $S \subseteq \boldsymbol{x}$ such that every variable in S branches completely in F. Observe that if we further restrict to the variables in S then the formula simplifies further and becomes a $\sum^2 \prod^2 \mathsf{ROF}$. Therefore, if S is large enough to apply the hardness of representation result from Lemma 26, the proof is complete. This is indeed the case whenever $|S| \geq 5$.

▶ **Lemma 27.** *Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Assume that both $(o)_L$ and $(o)_R$ are multiplication gates. Moreover, suppose there exists a set of variables S with $|S| \geq 5$ such that every $x_i \in S$ branches completely in F.*

If no nonzero polynomial $g \in \mathcal{C} \cup \mathcal{E}^2$ vanishes at $\mathbf{0}$, then $\mathcal{P}_n \nmid p_F$.

The second case is more delicate and relies on exploiting the structural constraints imposed by the definition of partial branching. The high level idea is that we first prove that if hardness of representation does not hold, i.e., that a high degree monomial divides p_F, then some variable must branch in one of the children of the root.

▶ **Lemma 28.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula, and suppose that both $(o)_L$ and $(o)_R$ are multiplication gates. If, for some $\boldsymbol{\alpha} \in \mathbb{F}^n$, we have $p_F(\boldsymbol{x} + \boldsymbol{\alpha}) \sim \mathcal{P}_n$, then there exists a variable $x_i \in \boldsymbol{x}$ that branches in F.*

Assume then that x_1 branches in $(o)_{\mathrm{L}}$. Since it does not completely branch, this means that in $(o)_{\mathrm{R}}$ it appears in exactly one child (i.e., a grandchild of the root). We exploit this structure to eventually prove that p_F cannot be divisible by high degree monomial.

▶ **Lemma 29.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that both $(o)_L$ and $(o)_R$ are multiplication gates, and that no variable branches completely in F. Moreover, Suppose that no polynomial in $\mathcal{A}^3_{F_{(o)_L}} \cup \mathcal{A}^3_{F_{(o)_R}}$ vanishes at $\mathbf{0}$. Then, if $n \geq 5$, we have $\mathcal{P}_n \nmid p_F$.*

In the third case, we present a procedure that attempts to reduce the setting to one of the two previous cases by applying generic assignments, to get rid of the few variables that completely branch. Note, however, that once we assign a value to a variable, we change the

structure of the formula, and new completely branched variables may appear. We show that if this is indeed the case, and it repeats itself through several iterations of fixing completely branched variables, then this outs a very rigid structure on the formula, which allows us to prove that the resulting polynomial cannot be divisible by $\mathcal{P}_n$.

▶ **Lemma 30.** *Let $F \in \sum^2 \mathsf{R2F}$ be a totally non-structural formula. Suppose that both $(o)_L$ and $(o)_R$ are multiplication gates. Let S_1 be the set of completely branched variables in F, and assume that $1 \leq |S_1| \leq 4$. Furthermore, assume that no nonzero polynomial in $\mathcal{C} \cup \mathcal{E}^3$ vanishes at $\mathbf{0}$.*

Then, if $n \geq 23$, we have $\mathcal{P}_n \nmid p_F$.

Combining all these results we obtain our hardness of representation result for the totally non-structural case.

▶ **Proposition 31.** *Let $F \in \sum^2 \mathsf{R2F}$ be totally non-structural. Suppose that no nonzero polynomial in $\mathcal{C} \cup \mathcal{E}^3$ vanishes at $\mathbf{0}$. Then, if $n \geq 23$, we have $\mathcal{P}_n \nmid p_F$.*

2.2 Proof overview of our whitebox algorithm (Theorem 3)

We follow the high-level approach of the whitebox algorithm introduced in [6]. Let S_4 denote the set of read-4 variables in a formula $F \in \mathsf{R4F}$. Our algorithm proceeds iteratively: in each iteration, it removes at least one variable from S_4 without introducing any new variables into it, while preserving nonzeroness of F. After at most n iterations, the resulting formula is an $\mathsf{R3F}$. We then verify its identity using the whitebox algorithm for $\mathsf{R3F}$s from [34].

To achieve this, we follow the cases handled in the proof of Theorem 2, and remove variables by following its arguments.

2.3 Proof overview of our orbit results

All of our orbit results follow from the simple observation that hardness of representation results for polynomial classes that are closed under translations and have low individual degree imply PIT not only for the backbone class[9], but also for the orbit class itself.

The following is an equivalent definition of hardness of representation, this is the notion that we primarily use in the full version.

▶ **Definition 32** (Hardness of Representation)**.** *A polynomial class $\mathcal{C} \subseteq \mathbb{F}[\boldsymbol{x}]$ is said to be m-*hard *if it is closed under $\mathbf{0}$-substitutions and no polynomial $p \in \mathcal{C}$ is divisible by a monomial with support size at least m.* ⌟

Assume we are concerned with an m-hard polynomial class $\mathcal{C}$. Then every polynomial $f(\boldsymbol{y}) \in \mathcal{C}$ contains a monomial whose support size is at most $m-1$. Since the action of every nonzero matrix $\boldsymbol{A} \in \mathrm{GL}_n(\mathbb{F})$ on $\mathbb{F}[\boldsymbol{y}]$ induces a degree-preserving isomorphism, if f has low degree, then $f(\boldsymbol{A}\boldsymbol{x})$ must contain a monomial of small support. We extend this observation to the action of $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ by first translating our polynomials by an appropriate vector of scalars.

This observation immediately lifts a large number of known results for backbone classes to their corresponding orbit classes.

[9] $\mathcal{C}$ denotes the backbone class corresponding to the orbit class $\mathcal{C}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$.

References

1. Manindra Agrawal. Proving lower bounds via pseudo-random generators. In Sundar Sarukkai and Sandeep Sen, editors, *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science*, pages 92–105, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. doi:10.1007/11590156_6.
2. Manindra Agrawal and Somenath Biswas. Primality and identity testing via chinese remaindering. *J. ACM*, 50(4):429–443, 2003. doi:10.1145/792538.792540.
3. Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Annals of Mathematics*, 160(2):781–793, September 2004. doi:10.4007/annals.2004.160.781.
4. Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.
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, December 2015. doi:10.1007/s00037-015-0097-4.
7. Robert Andrews. Algebraic hardness versus randomness in low characteristic. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 37:1–37:32. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.37.
8. Vishwas Bhargava and Sumanta Ghosh. Improved hitting set for orbit of roabps. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, pages 30–1. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.30.
9. Pranav Bisht, Nikhil Gupta, and Ilya Volkovich. Towards Identity Testing for Sums of Products of Read-Once and Multilinear Bounded-Read Formulae. In Patricia Bouyer and Srikanth Srinivasan, editors, *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023)*, volume 284 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:23, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2023.9.
10. Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5), 2018. doi:10.1145/3209663.
11. Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019. doi:10.4086/TOC.2019.V015A013.
12. Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, June 1978. doi:10.1016/0020-0190(78)90067-4.
13. Pranjal Dutta and Sumanta Ghosh. Sigact news complexity theory column 121. *SIGACT News*, 55(2):53–88, 2024. doi:10.1145/3674159.3674165.
14. Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM Journal on Computing*, 39(4):1279–1293, 2010. doi:10.1137/080735850.
15. Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Communications of the ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.
16. Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 163–172. ACM, 2012. doi:10.1145/2213977.2213995.

17 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *International Workshop on Approximation Algorithms for Combinatorial Optimization*, pages 527–542. Springer, 2013. doi:10.1007/978-3-642-40328-6_37.

18 Michael A. Forbes and Amir Shpilka. A pspace construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2018, pages 1180–1192, New York, NY, USA, 2018. Association for Computing Machinery. doi:10.1145/3188745.3188792.

19 Michael Andrew Forbes. *Polynomial identity testing of read-once oblivious algebraic branching programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

20 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness. *SIAM J. Comput.*, 51(2):315–335, 2022. doi:10.1137/20M1347395.

21 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth Three . In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 578–587, Los Alamitos, CA, USA, October 2013. IEEE Computer Society. doi:10.1109/FOCS.2013.68.

22 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complex.*, 26(4):835–880, 2017. doi:10.1007/S00037-016-0141-Z.

23 Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 821–830, 2017. doi:10.1145/3055399.3055440.

24 J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, pages 262–272, New York, NY, USA, 1980. Association for Computing Machinery. doi:10.1145/800141.804674.

25 Ivan Hu, Dieter van Melkebeek, and Andrew Morgan. Polynomial identity testing via evaluation of rational functions. *Theory of Computing*, 20(1):1–70, 2024. doi:10.4086/toc.2024.v020a001.

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

27 Nimrod Kaplan and Amir Shpilka. Polynomial identity testing for read-4 arithmetic formulas. *Electronic Colloquium on Computational Complexity (ECCC)*, page 95, 2026. URL: https://eccc.weizmann.ac.il/report/2026/076/.

28 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. doi:10.1016/J.TCS.2012.03.041.

29 Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019.

30 Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-Optimal Bootstrapping of Hitting Sets for Algebraic Models. *Theory Comput.*, 19:1–30, 2023. doi:10.4086/TOC.2023.V019A012.

31 László Lovász. On determinants, matchings and random algorithms. *Fundamentals of Computation Theory*, 79:565–574, January 1979.

32 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992. doi:10.1145/146585.146605.

33 Meena Mahajan, B. V. Raghavendra Rao, and Karteek Sreenivasaiah. Building above read-once polynomials: Identity testing and hardness of representation. *Algorithmica*, 76(4):890–909, December 2016. doi:10.1007/s00453-015-0101-z.

34 Meena Mahajan, B.V. Raghavendra Rao, and Karteek Sreenivasaiah. Monomials, multilinearity and identity testing in simple read-restricted circuits. *Theoretical Computer Science*, 524:90–102, 2014. doi:10.1016/j.tcs.2014.01.005.

35 Dori Medini and Amir Shpilka. Hitting sets and reconstruction for dense orbits in $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ circuits. *CoRR*, abs/2102.05632, 2021. arXiv:2102.05632.

36 Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Trans. Comput. Theory*, 10(3), May 2018. doi:10.1145/3196836.

37 Ketan Mulmuley. Geometric complexity theory v: Efficient algorithms for noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

38 Øystein Ore. Über höhere kongruenzen. *Norske Videnskaps-Akademi i Oslo. Forhandlinger (Proceedings of the Norwegian Academy of Science and Letters)*, 1922(12):1–8, 1922. Contains the original root bound for nonzero multivariate polynomials over finite fields, later used in polynomial identity testing (Schwartz–Zippel lemma).

39 Gautam Prakriya. Derandomizing isolation and polynomial identity testing. Doctoral thesis, 2019.

40 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005. doi:10.1007/S00037-005-0188-8.

41 Chandan Saha and Bhargav Thankey. Hitting sets for orbits of circuit classes and polynomial families. *ACM Trans. Comput. Theory*, 16(3), 2024. doi:10.1145/3665800.

42 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity, 2021. URL: https://github.com/dasarpmar/lowerbounds-survey.

43 Nitin Saxena. Progress on Polynomial Identity Testing. *Bull. EATCS*, 99:49–79, 2009.

44 Nitin Saxena. Progress on polynomial identity testing-II. *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146, 2014.

45 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. doi:10.1145/322217.322225.

46 Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, 1992. doi:10.1145/146585.146609.

47 Nadav Shamir. Polynomial identity testing for read-2 formulas and linear separation between read-3 and read-2 multilinear formulas. Master thesis, 2022.

48 Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In Irit Dinur, Klaus Jansen, Joseph Naor, and José Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 700–713, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. doi:10.1007/978-3-642-03685-9_52.

49 Amir Shpilka and Ilya Volkovich. On reconstruction and testing of read-once formulas. *Theory of Computing*, 10(18):465–514, 2014. doi:10.4086/toc.2014.v010a018.

50 Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. *Comput. Complex.*, 24(3):477–532, September 2015. doi:10.1007/S00037-015-0105-8.

51 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.

52 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.

53 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. doi:10.1016/J.IC.2014.09.004.

54 Richard Zippel. *Probabilistic algorithms for sparse polynomials*, pages 216–226. Springer Berlin Heidelberg, 1979. doi:10.1007/3-540-09519-5_73.