

Towards Identity Testing for Sums of Products of Read-Once and Multilinear Bounded-Read Formulae

Pranav Bisht  

Computer Science Department, Boston College, Chestnut Hill, MA, USA
Department of Computer Science and Engineering, IIT(ISM) Dhanbad, India

Nikhil Gupta   

Computer Science Department, Boston College, Chestnut Hill, MA, USA

Ilya Volkovich  

Computer Science Department, Boston College, Chestnut Hill, MA, USA

Abstract

An arithmetic formula is an arithmetic circuit where each gate has fan-out one. An *arithmetic read-once formula* (ROF in short) is an arithmetic formula where each input variable labels at most one leaf. In this paper we present several efficient blackbox *polynomial identity testing* (PIT) algorithms for some classes of polynomials related to read-once formulas. Namely, for polynomial of the form:

- $f = \Phi_1 \cdot \dots \cdot \Phi_m + \Psi_1 \cdot \dots \cdot \Psi_r$, where Φ_i, Ψ_j are ROFs for every $i \in [m], j \in [r]$.
- $f = \Phi_1^{e_1} + \Phi_2^{e_2} + \Phi_3^{e_3}$, where each Φ_i is an ROF and e_i -s are arbitrary positive integers.

Earlier, only a whitebox polynomial-time algorithm was known for the former class by Mahajan, Rao and Sreenivasaiah (Algorithmica 2016).

In the same paper, they also posed an open problem to come up with an efficient PIT algorithm for the class of polynomials of the form $f = \Phi_1^{e_1} + \Phi_2^{e_2} + \dots + \Phi_k^{e_k}$, where each Φ_i is an ROF and k is some constant. Our second result answers this partially by giving a polynomial-time algorithm when $k = 3$. More generally, when each Φ_1, Φ_2, Φ_3 is a multilinear bounded-read formulae, we also give a quasi-polynomial-time blackbox PIT algorithm.

Our main technique relies on the *hardness of representation* approach introduced in Shpilka and Volkovich (Computational Complexity 2015). Specifically, we show hardness of representation for the resultant polynomial of two ROFs in our first result. For our second result, we lift hardness of representation for a sum of three ROFs to sum of their powers.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Identity Testing, Derandomization, Bounded-Read Formulae, Arithmetic Formulas

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2023.9

Related Version *Full Version:* <https://eccc.weizmann.ac.il/report/2023/109/>

Acknowledgements The authors would like to thank the anonymous referees for their detailed comments and suggestions on the previous version of the paper.

1 Introduction

Polynomial Identity Testing (PIT) is a central problem in the area of algebraic complexity theory. Given a multivariate polynomial in the form of an arithmetic circuit or a formula Φ , one is asked to decide whether Φ computes the identically zero polynomial, i.e. every coefficient in the monomial expansion of the polynomial computed by Φ is zero. There are



© Pranav Bisht, Nikhil Gupta, and Ilya Volkovich;
licensed under Creative Commons License CC-BY 4.0

43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2023).

Editors: Patricia Bouyer and Srikanth Srinivasan; Article No. 9; pp. 9:1–9:23



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

two types of PIT algorithms – whitebox and blackbox. In the former, one can look inside the circuit or formula while in the latter one can only access evaluations of the formula on field points of their choice.

PIT is one of the important problems in the class BPP (actually in co-RP) for which a polynomial-time deterministic algorithm is yet to be found. The blackbox randomized algorithm for PIT is extremely simple: given an input polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ of degree d , pick any set $S \subseteq \mathbb{F}$ of size greater than d and evaluate f on a random point sampled from S^n . Declare the polynomial to be an identity if the evaluation is zero and a non-identity otherwise. If the polynomial was actually zero then this algorithm cannot err, otherwise for a non-zero input, this random evaluation can be zero with probability at most $d/|S|$ by the *Schwartz-Zippel-DeMillo-Lipton* Lemma [50, 57, 16].

Derandomizing PIT is intimately tied to proving circuit lower bounds. A deterministic sub-exponential-time PIT algorithm yields either a super-polynomial Boolean or a super-polynomial arithmetic circuit lower bound [29, 28, 1]. Conversely, a super-polynomial arithmetic circuit lower bound implies a deterministic sub-exponential time PIT algorithm [29]. We refer the reader to the excellent survey [36] for a detailed exposition on this *hardness vs randomness* trade-off in the algebraic setting. PIT also finds applications in the problems of primality testing [2] and finding perfect matchings in graphs [38].

An arithmetic formula is an arithmetic circuit whose underlying graph is a tree. While derandomizing PIT for arithmetic formulae is still open, various interesting restricted classes have found efficient deterministic PIT algorithms. One such natural restriction is to bound the number of times a variable can appear in a formula. An arithmetic read-once formula (ROF in short) is a formula where each variable appears at most once. Shpilka and Volkovich considered the more general class of sum of k ROFs, where k is some constant. They devised a quasi-polynomial-time deterministic algorithm for this class in [53], which was later improved to polynomial time in [41]. An even more generalized model is a read- k formula, where every variable can appear at most k times, for some constant k . For the class of multilinear read- k formulas, [5] give a deterministic quasi-polynomial-time PIT algorithm. Note that the class of sum of k ROFs forms a strict subclass of multilinear read- k formulas.

From a single ROF to a sum of ROFs, the next model to consider is sum-of-products of ROFs. More generally, let \mathcal{C} be any natural circuit class like ROFs, then one can define the class $\sum^{[k]} \prod \mathcal{C}$ which consists of polynomials of the form $f = \sum_{i=1}^k \prod_{j=1}^m f_{ij}$, where each f_{ij} belongs to the class \mathcal{C} . One can also define its sub-class $\sum^{[k]} \wedge \mathcal{C}$ where each product gate takes the same input. Namely, the class consists of polynomials of the form $f = \sum_{i=1}^k f_i^{e_i}$, where each $f_i \in \mathcal{C}$. The work of [44] proved lower bounds against the class $\sum^{[k]} \prod$ ROF, when k is constant and the product gates have certain fan-in restriction. For PIT, [39] designed a polynomial-time *whitebox* algorithm for the sub-class $\sum^{[2]} \prod$ ROF. In this work, we give a polynomial-time *blackbox* PIT algorithm for this model. For a constant k , PIT for the class $\sum^{[k]} \wedge$ ROF was left as an open problem by [39]. Here, we give a polynomial-time (quasi-polynomial-time) blackbox PIT algorithm for $\sum^{[k]} \wedge \mathcal{C}$, when $k = 3$ and \mathcal{C} is the class of read-once (multilinear constant-read) formulas.

1.1 Motivations and Related Works

PIT for $\sum^{[k]} \prod \mathcal{C}$. One of the important results in the PIT literature is an efficient deterministic PIT algorithm for the class $\sum^{[k]} \prod \sum$, both in blackbox and whitebox setting, where k is a constant. The first subexponential PIT algorithm was given in [18]. The algorithm was in the whitebox setting and had quasi-polynomial run-time. Later, in [34], the

result was improved by presenting a polynomial-time whitebox algorithm. This was followed by a long line of work [32, 33, 47, 48, 49] which culminated in a polynomial-time blackbox algorithm.

A $\sum^{[k]} \prod \sum$ circuit over a field \mathbb{F} computes polynomials of the kind $\sum_{i=1}^k \prod_{j=1}^{d_i} \ell_{i,j}$, where $d_i \in \mathbb{N}$ for every $i \in [k]$ and $\ell_{i,j} \in \mathbb{F}[x_1, x_2, \dots, x_n]$ are linear polynomials for every $i \in [k], j \in [d_i]$. One natural way to extend this result is to replace $\ell_{i,j}$'s with more general arithmetic circuits, for which efficient deterministic PIT algorithms are known. Some interesting candidates for such circuits are sparse polynomials (or $\sum \prod$ circuits), ROFs, multilinear bounded-read formulae, etc. Clearly, each of these circuit classes subsume the class of linear polynomials over \mathbb{F} . Polynomial-time deterministic blackbox PIT algorithms are known for the classes of sparse polynomials [35] and ROFs [41], and the class of multilinear bounded-read formulae admits a quasi-polynomial-time blackbox PIT algorithm [5].

PIT for the class $\sum^{[k]} \prod \sum \prod$ is well-studied: Polynomial-time deterministic blackbox PIT algorithms are known for (syntactically) multilinear $\sum^{[k]} \prod \sum \prod$ circuits [46], for *constant-read* $\sum \prod \sum \prod$ circuits [3, 7], and for the class $\sum^{[3]} \prod \sum \prod^{[2]}$ [43]; and a quasi-polynomial-time PIT algorithm for $\sum^{[k]} \prod \sum \prod^{[\delta]}$ was given in [17], where δ is also a constant. A deterministic sub-exponential PIT was given for the class $\sum \prod \sum \prod$ in the breakthrough result of [37]¹. Note that there is no top fan-in restriction in their result. However a polynomial-time PIT algorithm continues to be elusive.

In this work, we explore the other route, i.e., in the direction of $\sum^{[k]} \prod$ ROF, which consists of circuits of the kind $\sum_{i=1}^k \prod_{j=1}^{d_i} \Phi_{i,j}$, where every $\Phi_{i,j}$ is an ROF over \mathbb{F} . The class of ROFs has been studied extensively in the Boolean as well as algebraic worlds. The results in the Boolean world include learning algorithms for Boolean ROFs and some structural properties of Boolean read-once functions [6, 30, 13, 14]. In the arithmetic world, we have the following results for the class of ROFs: A deterministic polynomial-time blackbox PIT algorithm [53, 41], efficient reconstruction algorithms [27, 12, 11, 10, 52, 41], quasi-polynomial-time deterministic blackbox PIT algorithms for the *orbit*² of ROFs [40, 45], a randomized polynomial-time reconstruction algorithm for orbits of ROFs [24], and characterization of read-once polynomials [56]. The investigation of PIT for $\sum^{[k]} \prod$ ROF might lead to the discovery of new ideas and techniques, which may be helpful in approaching PIT for general arithmetic circuits and formulae.

A polynomial-time deterministic whitebox PIT algorithm for $\sum^{[2]} \prod$ ROF was given in [39]. In this work, we give a polynomial-time deterministic blackbox PIT algorithm for $\sum^{[2]} \prod$ ROF (see Theorem 1). Our algorithm works over any field satisfying some mild condition on its size. The blackbox nature makes the problem quite non-trivial and we introduce a new tool for handling this: *0-irreducibility* (for more details, see Definition 5 and Observation 7). This tool could also be crucial to obtain PIT algorithms for $\sum^{[k]} \prod$ ROF, where $k \geq 3$ is a constant and for other interesting circuit classes.

PIT for $\sum^{[k]} \wedge \mathcal{C}$. The circuit class $\sum^{[k]} \wedge \mathcal{C}$ consists of arithmetic circuits of the type $\Phi_1^{e_1} + \dots + \Phi_k^{e_k}$, where $\Phi_1, \dots, \Phi_k \in \mathcal{C}$ and $e_1, \dots, e_k \in \mathbb{N}$ are arbitrary. Apart from being a natural and interesting problem in itself, developing efficient PIT algorithm for this class is also important from the viewpoint of PIT for the class $\sum^{[k]} \prod \mathcal{C}$, which subsumes $\sum^{[k]} \wedge \mathcal{C}$.

¹ [37] gave a much more general result, which solves PIT for any bounded-depth arithmetic circuit in sub-exponential time.

² Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$. Then, the orbit of f is the set of polynomials $f(Ax)$, where A varies over all $n \times n$ invertible matrices over \mathbb{F} .

Another reason for studying PIT algorithms for $\sum^{[k]} \wedge \mathcal{C}$ is that it generalizes the PIT for $\sum^{[k]} \mathcal{C}$, which is comprised of the circuits $\Phi_1 + \dots + \Phi_k$, where $\Phi_1, \dots, \Phi_k \in \mathcal{C}$. In this work, we instantiate \mathcal{C} with the classes of ROFs and multilinear bounded-read arithmetic formulae, and take k to be equal to 3 (see Theorems 2 and 3). For the sake of discussion, let \mathcal{C}_k be the class of read- k arithmetic formulae over a field \mathbb{F} . Although, $\sum^{[3]} \wedge \text{ROF}$ is contained in $\sum^{[3]} \wedge \mathcal{C}_k$, the reason for mentioning them separately is that in the case of ROFs, we obtain a deterministic polynomial-time blackbox PIT, whereas the time complexity of the blackbox PIT in the case of multilinear bounded-read formulae is quasi-polynomial.

A deterministic polynomial-time PIT algorithm is known for the class $\sum^{[k]} \text{ROF}$ [53, 41], which was built over the efficient PIT algorithm for the class of (single) ROFs [41]. This is a non-trivial generalization because the class of ROFs is not closed with respect to addition of ROFs. Now, the next level of generalization is to allow arbitrary powers of ROFs in the sum of k ROFs. This brings us to the class $\sum^{[k]} \wedge \text{ROF}$, for which obtaining efficient PIT has been mentioned as an open problem in [39]. PIT for the “bounded-depth variant” of this class has been studied: A polynomial-time blackbox PIT algorithm is given in [3] for the class of sum of powers of constantly many bounded-depth ROFs³. This algorithm is based on carefully exploiting the *Jacobian* of such circuits and the polynomial running time of their PIT crucially depends on the “bounded-depth nature” of the underlying formulae. The story of the class $\sum^{[k]} \wedge \mathcal{C}_k$ is also similar. [3] gives a polynomial-time deterministic blackbox PIT for the “bounded-depth” variant of this class. However, it is not clear to us how to extend their techniques to obtain a polynomial-time PIT for the classes $\sum^{[k]} \wedge \text{ROF}$ and $\sum^{[k]} \wedge \mathcal{C}_k$ in the arbitrary-depth setting.

1.2 Our Results

Now we start with our main results. Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be an input polynomial of degree d . Our results below hold over *any* field \mathbb{F} satisfying $|\mathbb{F}| > n \cdot d$. Otherwise, we assume to have access to a sufficiently large extension field. We note that the requirement for large enough field size is intrinsically necessary for any *blackbox* PIT algorithm.

Our first result is a blackbox PIT algorithm for $\sum^{[2]} \prod \text{ROF}$. It improves a previous result of [39], which gave a polynomial-time whitebox PIT for the same model.

► **Theorem 1** (Blackbox PIT for $\sum^{[2]} \prod \text{ROF}$). *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a polynomial of degree at most d computed as $f = \Phi_1 \cdots \Phi_m + \Psi_1 \cdots \Psi_r$, where $\Phi_1, \dots, \Phi_m, \Psi_1, \dots, \Psi_r$ are ROFs. Then there exists a deterministic algorithm that given blackbox access to f decides whether f is identically zero, in time $\text{poly}(n, d)$.*

► **Remark 1.** The parameters m and r used in the above theorem can be arbitrary as long as the degree of the polynomial computed by $\Phi_1 \cdots \Phi_m + \Psi_1 \cdots \Psi_r$ is at most d .

Due to limitations of space, the proof of this theorem is given in Appendix A. It is based on the high-level proof overview given in Section 1.3.

In the following two theorems we give blackbox PIT algorithms for the classes $\sum^{[3]} \wedge \text{ROF}$ and $\sum^{[3]} \wedge \mathcal{C}_k$, where \mathcal{C}_k is the class of multilinear read- k arithmetic formulae (Definition 40). Although ROFs are subsumed by multilinear bounded-read formulae, we are stating different results for them since we obtain a polynomial-time PIT for $\sum^{[3]} \wedge \text{ROF}$, whereas the runtime

³ In terminology of [3], such formulae are called as sum of constantly many bounded-depth *occur-once* formulae. In fact, a more general result along with other results was given in [3] - a polynomial-time deterministic blackbox PIT algorithm for the class of bounded-depth *bounded-occur* arithmetic formulae.

for the PIT algorithm for $\sum^{[3]} \wedge \mathcal{C}_k$ is quasi-polynomial. An interesting common thread in these results is that the time complexity of the blackbox PIT algorithm for $\sum^{[3]} \wedge \text{ROF}$ (similarly, $\sum^{[3]} \wedge \mathcal{C}_k$) is same as the blackbox PIT algorithm for $\sum^{[3]} \text{ROF}$ (respectively, $\sum^{[3]} \mathcal{C}_k$), which is strictly weaker than $\sum^{[3]} \wedge \text{ROF}$ (respectively, $\sum^{[3]} \wedge \mathcal{C}_k$).

► **Theorem 2** (Blackbox PIT for $\sum^{[3]} \wedge \text{ROF}$). *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a polynomial of degree at most d computed as $f = \Phi_1^{e_1} + \Phi_2^{e_2} + \Phi_3^{e_3}$ where Φ_1, Φ_2, Φ_3 are ROFs and $e_1, e_2, e_3 \in \mathbb{N}$. Then there exists a deterministic algorithm that given blackbox access to f decides whether $f \equiv 0$, in time $\text{poly}(n, d)$.*

► **Theorem 3** (Blackbox PIT for $\sum^{[3]} \wedge \mathcal{C}_k$). *Let $k \in \mathbb{N}$ be a constant and let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a polynomial of degree at most d computed as $f = \Phi_1^{e_1} + \Phi_2^{e_2} + \Phi_3^{e_3}$, where Φ_1, Φ_2, Φ_3 are multilinear read- k arithmetic formulae and $e_1, e_2, e_3 \in \mathbb{N}$. Then there exists a deterministic algorithm that given blackbox access to f decides whether $f \equiv 0$, in time $(n \cdot d)^{O(\log n)}$.*

Due to limitations of space, the proofs of these two theorems are given in Appendix B. In fact, we prove a more general result in Appendix B (see Theorem 58), which subsumes Theorems 2 and 3. For simplicity of exposition, we give a high-level proof overview of Theorem 2 in Section 1.3.2. The proof overview of Theorem 3 is exactly on the same line as that of Theorem 2.

1.3 Proof Overview and Techniques

In this section, we give the high level overviews of the proofs of Theorems 1, 2, and 3. The underlying theme of these proofs is the *hardness of representation* approach, which was first introduced in [53], where PIT algorithms for sums of constantly many ROFs were given. In its initial avatar, hardness of representation was given for sum of constantly many read-once polynomials (ROPs)⁴ satisfying some technical conditions referred to as “**0**-justified” (Definition 2). Formally, let \mathbb{F} be an arbitrary field and $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be **0**-justified ROPs. Set $A \triangleq A_1 + \dots + A_k \neq 0$. Then for $n \geq 3k$ the monomial $x_1 \cdots x_n$ does not divide A . Hardness of representation approach also sits at the core of the PIT algorithm for the class of multilinear bounded-read arithmetic formulae given in [5]. In this paper, we work with a more general form of this approach. See Definition 25 and Fact 27 in this regard.

There is also an alternate way to view hardness of representation, which is popularly called *low-support concentration* in literature [4, 21, 20, 19, 26, 25, 54]. The idea is to choose a “nice” point $(a_1, \dots, a_n) \in \mathbb{F}^n$ for an input polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ such that the *shifted* polynomial $f(x_1 + a_1, \dots, x_n + a_n)$ has a non-zero monomial of low *support-size* (number of variables appearing in the monomial). For a polynomial with such a low-support monomial, efficient blackbox PIT is known (Fact 22). In this work, we shift by a *justifying assignment* or an *irreducibility-preserving assignment* (see Definitions 2, 5) in order to achieve hardness of representation, which in turn proves existence of a low-support monomial (Fact 27).

1.3.1 Proof Overview of Theorem 1

In Theorem 1, we give a polynomial-time blackbox PIT for $\sum^{[2]} \amalg \text{ROF}$, which consists of polynomials of the kind $f = A_1 \cdots A_m + B_1 \cdots B_r$, where every $A_\ell, B_t \in \mathbb{F}[x_1, \dots, x_n]$ are read-once polynomials (ROPs), i.e., the polynomials computed by ROFs (Definition 28).

⁴ A *read-once polynomial* is a polynomial that is computable by a read-once formula

We are given blackbox access to such an $f = A_1 \cdots A_m + B_1 \cdots B_r$, where $\deg(f) \leq d$ and we want to determine whether $f \equiv 0$ or not. To accomplish this, we design a *hitting-set generator*, that is a polynomial map $\mathcal{G} = (\mathcal{G}^1, \dots, \mathcal{G}^n) : \mathbb{F}^w \rightarrow \mathbb{F}^n$ which preserves non-zerosness, formally $f(x_1, \dots, x_n) \equiv 0$ if and only if $f(\mathcal{G}^1, \dots, \mathcal{G}^n) \equiv 0$, where w is a constant and $\max\{\deg(\mathcal{G}^i) : i \in [n]\} \leq \delta$. Then, $f(\mathcal{G})$ becomes a w -variate polynomial, which has degree at most $d \cdot \delta$. Since w is a constant, it is easy to test the zeroness of $f(\mathcal{G})$ in time $(nd\delta)^{O(w)}$. The map \mathcal{G} in our case is the generator $G_{n,4}$ given in Definition 20, with $w = 8$.

Now, let us see why $G_{n,4}$ is a correct generator for the class $\sum^{[2]} \prod$ ROF. Recall $f = A_1 \cdots A_m + B_1 \cdots B_r$. As every A_ℓ, B_t are ROPs, we can assume without loss of generality that they are irreducible (see Fact 29). We now apply the standard trick of *simplifying* the polynomial. Formally, let $g \triangleq \gcd(A_1 \cdots A_m, B_1 \cdots B_r)$ and $f' \triangleq \frac{f}{g}$. Then, we can write $f = g \cdot f'$, where g is called the *simple* part of f . Since g is a product of non-zero ROPs (see Fact 29), it follows from a result of [41] (see Fact 38) and the *multiplicative property* of a generator that $f(G_{n,4}) \equiv 0$ if and only if $f'(G_{n,4}) \equiv 0$. So, we can assume without loss of generality that $f' = f = A_1 \cdots A_m + B_1 \cdots B_r$. Then, there are two possibilities: Either $f \in \mathbb{F}$ or for every $\ell \in [m], t \in [r]$, A_ℓ and B_t are co-prime. The first case is trivial. Now, we talk about the second case.

Fix $A = A_\ell$ and $B = B_t$ arbitrarily. Since $\gcd(A, B) = 1$, if x appears in A then the resultant of A and B with respect to x , denoted $\text{Res}_x(A, B)$, is a non-zero polynomial (see Definition 9 and Fact 10). As A, B are ROPs, they are multilinear, and can be written as $A = A'_1 x + A'_0$ and $B = B'_1 x + B'_0$, where $A'_1, A'_0, B'_1, B'_0 \in \mathbb{F}[\mathbf{x} \setminus \{x\}]$. Then, it follows from Definition 9 that $\text{Res}_x(A, B) = A'_1 B'_0 - A'_0 B'_1$. Since A and B are co-prime polynomials, $\text{Res}_x(A, B) \neq 0$. If we have a generator \mathcal{G} that hits this resultant, then we are done as $A(\mathcal{G})$ will be co-prime to every $B(\mathcal{G})$, which certifies that $f(\mathcal{G}) \neq 0$. This approach has been utilized earlier also and is formally stated in Fact 24. In order to argue that $f(G_{n,4}) \neq 0$, it suffices to prove that $G_{n,3}$ hits the resultant $\text{Res}_x(A, B)$, i.e. $(\text{Res}_x(A, B))(G_{n,3}) \neq 0$.

Now, we argue that $(\text{Res}_x(A, B))(G_{n,3}) \neq 0$. For this, we introduce the notion of *zero-irreducibility*. We say that a polynomial $g \in \mathbb{F}[x_1, \dots, x_n]$ is **0-irreducible**, if for every proper subset $I \subsetneq [n]$, the restricted polynomial $g|_{\mathbf{x}_I = \mathbf{0}_I}$ is irreducible and $g(\mathbf{0}) \neq 0$ (Definition 5). Let us first see why **0-irreducible** ROPs are interesting in this scenario. Let \tilde{A} and \tilde{B} be two **0-irreducible** ROPs and $x \in \text{var}(\tilde{A})$. We show that there exists a monomial in $\text{Res}_x(\tilde{A}, \tilde{B})$, which has at most two variables (see Corollary 46). This along with Fact 22 implies that $(\text{Res}_x(\tilde{A}, \tilde{B}))(G_{n,2}) \neq 0$. To show that a monomial of support at most two exists in $\text{Res}_x(\tilde{A}, \tilde{B})$, we prove a *hardness of representation* theorem for the resultant of two **0-irreducible** ROPs. In particular, we show that if \tilde{A}, \tilde{B} are **0-irreducible** ROPs then there do not exist three distinct variables x_1, x_2, x_3 such that $x_1 x_2 x_3$ divides $\text{Res}_x(\tilde{A}, \tilde{B})$ (see Lemma 45). This result is the heart of the proof of Theorem 1.

Now let us see how to transform the original irreducible ROPs A, B to **0-irreducible** ROPs \tilde{A}, \tilde{B} . We show that there exists an assignment \mathbf{a} in the image of $G_{n,1}$ such that $\tilde{A} \triangleq A(\mathbf{x} + \mathbf{a})$ and $\tilde{B} \triangleq B(\mathbf{x} + \mathbf{a})$ are **0-irreducible** ROPs. For this, we need a tool called the *commutator* of a polynomial $g \in \mathbb{F}[x_1, x_2, \dots, x_n]$ (Definition 13), denoted $\Delta_{i,j}g$, where $i, j \in [n]$. Since A, B are irreducible multilinear polynomials, all the commutators of A and B are non-zero (Corollary 17). We show that if we have an $\mathbf{a} \in \mathbb{F}^n$ such that $A(\mathbf{a}) \neq 0, B(\mathbf{a}) \neq 0$ and for every $i \neq j \in [n]$, $(\Delta_{i,j}A)(\mathbf{a}) \neq 0, (\Delta_{i,j}B)(\mathbf{a}) \neq 0$, then $A(\mathbf{x} + \mathbf{a})$ and $B(\mathbf{x} + \mathbf{a})$ are **0-irreducible** ROPs. The nice structure of a commutator of an ROP given in Corollary 34 turns out to be extremely helpful in showing that the desired tuple \mathbf{a} is in the image of $G_{n,1}$.

On putting the things together, we get that $(\text{Res}_x(A, B))(G_{n,2} + G_{n,1}) \neq 0$. Since $G_{n,3} = G_{n,2} + G_{n,1}$ (see Fact 21), we have $\text{Res}_x(A, B)(G_{n,3}) \neq 0$.

1.3.2 Proof Overview of Theorems 2 and 3

In Section B, we prove a result (see Theorem 58) which captures both Theorems 2 and 3. However, for the sake of keeping the discussion simple and yet deliver the main ideas, we restrict ourselves to ROFs. In particular, we give a high-level proof overview of Theorem 2.

We are given blackbox access to an f computed by a circuit in $\sum^{[3]} \wedge$ ROF and we want to determine whether f is zero or not. Then, there exist three ROPs $A, B, R \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and $e_1, e_2, e_3 \in \mathbb{N}$ such that $f = A^{e_1} + B^{e_2} + R^{e_3}$. We prove that $f \equiv 0$ if and only if $f(G_{n,10}) \equiv 0$, where $G_{n,10}$ is given in Definition 20. The main crux of this result is the hardness of representation for f . We show that if an assignment $\mathbf{a} \in \mathbb{F}^n$ is a common *justifying assignment* (Definition 2) of A, B, R then $f(\mathbf{x} + \mathbf{a})$ is either zero or for every set $J \subseteq [n]$ of size 10, $f' \triangleq f(\mathbf{x} + \mathbf{a})$ is not divisible by the monomial $\prod_{j \in J} x_j$. This hardness of representation then implies existence of a monomial of support-size at most 9 in f' and therefore $G_{n,9}$ hits f' (see Fact 27). Formally, $f' \equiv 0$ if and only if $f'(G_{n,9}) \equiv 0$. We know from [53] that such an \mathbf{a} is in the image of $G_{n,1}$ (see Fact 30). Since $G_{n,10} = G_{n,9} + G_{n,1}$ (see Fact 21), we get that $G_{n,10}$ is a generator for f . Now, it is not difficult to argue that given blackbox access to f , we can test whether f is zero or not in $\text{poly}(n, d)$ -time, where $d = \deg(f)$.

The hardness of representation theorem mentioned above crucially uses the fact that f is a sum of powers of three ROFs. The proof proceeds by analyzing various cases originating from the comparison of the parameters e_1, e_2 , and e_3 mentioned above. Here, we assume without loss of generality that $e_1 \geq e_2 \geq e_3$. If $e_1 > e_2$ then it is easy to show the required hardness of representation result. A major chunk of the proof is devoted to analyze the case when $e_1 = e_2 = e$. In this part, the following factorization becomes pivotal.

$$A^e - B^e = \prod_{\ell \in [e]} (A - \omega^\ell B),$$

where ω is a primitive e -th root of unity. It may seem from here that our result only holds over fields that contain ω . However, it is not the case. We show that it is possible to “massage” e in such a way that a primitive e -th root of unity is always present in the underlying field (or an appropriate extension). Our proof crucially exploits the following two properties of ROFs: 1) the class of ROFs is closed under product of variable disjoint ROFs, and 2) the hardness of representation result for the sum of any three $\mathbf{0}$ -justified ROFs given in [53] (see Fact 35).

2 Preliminaries

For a field \mathbb{F} , its algebraic closure is denoted as $\overline{\mathbb{F}}$. \mathbb{N} represents the set of natural numbers. For $n \in \mathbb{N}$, $[n] \triangleq \{1, \dots, n\}$. Unless otherwise specified, $\mathbf{x} \triangleq \{x_1, \dots, x_n\}$. We denote the sets of variables by $\mathbf{x}, \mathbf{y}, \mathbf{z}$; polynomials by $f, g, h, u, v, A, B, F, R$; elements of \mathbb{F} by α, β, a, b ; vectors over \mathbb{F} by \mathbf{a}, \mathbf{b} ; circuit classes by upper case calligraphic letters like \mathcal{C} ; and sets by I, J, K, L . For a polynomial $f \in \mathbb{F}[\mathbf{x}]$, we denote a monomial $x_1^{e_1} \cdots x_n^{e_n}$ in f by \mathbf{x}^e and for some $i \in [n]$, $\deg_{x_i}(f)$ denotes the degree of variable x_i in f when it is viewed as a polynomial in x_i over $\mathbb{F}[\mathbf{x} \setminus \{x_i\}]$. The *individual degree* of f is defined as $\max_{i \in [n]} \{\deg_{x_i}(f)\}$. A polynomial f is called *multilinear* if its individual degree is at most one. We define support of a monomial by $\text{supp}(\mathbf{x}^e) \triangleq \{i \in [n] \mid e_i > 0\}$ and denote support-size by $|\text{supp}(\mathbf{x}^e)|$.

We call $f, g \in \mathbb{F}[\mathbf{x}]$ are *similar*, denoted $f \sim g$, if there exists a non-zero $\alpha \in \mathbb{F}$ such that $f = \alpha \cdot g$. For a polynomial $f \in \mathbb{F}[\mathbf{x}]$ and a vector $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{F}^n$, the shifted polynomial is $f(\mathbf{x} + \mathbf{a}) \triangleq f(x_1 + a_1, \dots, x_n + a_n)$. We say that $f \in \mathbb{F}[\mathbf{x}]$ depends on x_i

if there exist $\mathbf{a} = (a_1, \dots, a_n) \in \mathbb{F}^n$ and $b \in \mathbb{F}$ such that $f(a_1, \dots, a_{i-1}, a_i, a_{i+1}, \dots, a_n) \neq f(a_1, \dots, a_{i-1}, b, a_{i+1}, \dots, a_n)$. Further, $\text{var}(f) \triangleq \{i \in [n] : f \text{ depends on } x_i\}$. Let $f \in \mathbb{F}[\mathbf{x}]$, $I \subseteq [n]$, and $\mathbf{a} \in \mathbb{F}^n$. Then, $f|_{\mathbf{x}_I=\mathbf{a}_I}$ is obtained by substituting $x_i = a_i$ in f for every $i \in I$. Clearly, $\text{var}(f|_{\mathbf{x}_I=\mathbf{a}_I}) \subseteq \text{var}(f) \setminus I$. Observe that this containment can be strict. For example, let $f = x_1x_2 + 1$, $\mathbf{a} = (0, 0)$, and $I = \{1\}$. Then, $\text{var}(f|_{\mathbf{x}_I=\mathbf{a}_I}) \subsetneq \text{var}(f) \setminus \{1\}$ as after setting $x_1 = 0$ in f , it no longer depends on x_2 . We are interested in those assignments where such undesirable losses do not happen. Such assignments, known as *justifying assignments*, have been earlier considered in [27, 12, 53]. Consider the following definition, which has been obtained by adding Property 2 to the definition of a justifying assignment given in [53, 52]. This modification has been made to suit our purpose.

► **Definition 2** (Justifying assignment). *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and $\mathbf{a} \in \mathbb{F}^n$. Then, \mathbf{a} is called a justifying assignment of f (equivalently, f is said to be \mathbf{a} -justified) if*

1. for every $I \subseteq \text{var}(f)$, $\text{var}(f|_{\mathbf{x}_I=\mathbf{a}_I}) = \text{var}(f) \setminus I$ and
2. $f(\mathbf{a}) \neq 0$.

► **Remark 3.** By convention, the identically zero polynomial is \mathbf{a} -justified for every $\mathbf{a} \in \mathbb{F}^n$.

The following nice property of a justifying assignment is implied by Proposition 2.3 of [53].

► **Fact 4.** *An assignment $\mathbf{a} \in \mathbb{F}^n$ is a justifying assignment of a polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ if and only if the condition given in Property 1 of Definition 2 holds for every $I \subseteq \text{var}(f)$ of size $|\text{var}(f)| - 1$ and $f(\mathbf{a}) \neq 0$.*

An $f \in \mathbb{F}[\mathbf{x}] \setminus \mathbb{F}$ is *irreducible* if it can not be written as a product of two non-constant polynomials in $\mathbb{F}[\mathbf{x}]$. Otherwise, f is reducible. By convention, every element of \mathbb{F} is reducible.

► **Definition 5** (Irreducibility preserving assignment). *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and $\mathbf{a} \in \mathbb{F}^n$. Then, \mathbf{a} is called an irreducibility preserving assignment of f if for every proper subset $I \subsetneq \text{var}(f)$, the restricted polynomial $f|_{\mathbf{x}_I=\mathbf{a}_I}$ is irreducible and $f(\mathbf{a}) \neq 0$. Equivalently, we say that f is \mathbf{a} -irreducible.*

For example, let $f = x_1 + x_2 + x_3$ and $\mathbf{a} = (0, 0, 1)$. Then, f is \mathbf{a} -irreducible over every field. Observe that if $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ is \mathbf{a} -irreducible for any $\mathbf{a} \in \mathbb{F}^n$ then f is irreducible over \mathbb{F} . Claim 6 below shows that irreducibility preserving assignments capture justifying assignments of irreducible polynomials. Note that the converse of this claim is not true. For example, let $f = (x_1 + 1)(x_2 + 1) + x_3$ and $\mathbf{a} = (0, 0, 0)$. Then, f is irreducible, \mathbf{a} -justified but is not \mathbf{a} -irreducible. Thus, for irreducible polynomials, the notion of irreducible preserving assignment is strictly stronger than that of justifying assignment.

▷ **Claim 6** ([8]). Let $f \in \mathbb{F}[\mathbf{x}]$ and $\mathbf{a} \in \mathbb{F}^n$. If f is \mathbf{a} -irreducible then f is \mathbf{a} -justified.

► **Observation 7.** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and $\mathbf{a} \in \mathbb{F}^n$. Then, f is \mathbf{a} -irreducible if and only if $f(\mathbf{x} + \mathbf{a})$ is $\mathbf{0}$ -irreducible.*

2.1 Basic Mathematical Facts

► **Fact 8** (Gauss Lemma). *Let \mathbb{F} be a field, $f \not\equiv 0 \in \mathbb{F}[\mathbf{x}, y]$, and $g \in \mathbb{F}[\mathbf{x}]$. Suppose $f|_{y=g(\mathbf{x})} \equiv 0$. Then, $y - g(\mathbf{x})$ is an irreducible factor of f .*

2.3 Commutator

This section is devoted to commutators of polynomials. This tool was defined in [51], where it was used in the context of polynomial factorization. It also played a crucial role in the deterministic reconstruction algorithm for read-once formulas (Definition 28) given in [52]. The following definition of a commutator of a polynomial is taken from [52]. This can be seen as a determinant of a special case Nisan’s partial derivative matrix.

► **Definition 13 (Commutator).** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and $i \neq j \in [n]$. Then, the commutator of f with respect to x_i and x_j , denoted $\Delta_{i,j}f$, is defined as*

$$\Delta_{i,j}f = f|_{x_i=1, x_j=1} \cdot f|_{x_i=0, x_j=0} - f|_{x_i=1, x_j=0} \cdot f|_{x_i=0, x_j=1}.$$

We note that this definition of a commutator of a polynomial is different from the definition given in [51]. However, it is not difficult to show that both these definitions have same properties for multilinear polynomials. We now note some useful properties of commutators of multilinear polynomials.

► **Observation 14.** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a multilinear polynomial and $i, j \in [n]$. Then, f can be written as $f = f_{i,j}x_i x_j + f_i x_i + f_j x_j + f_0$, where $f_{i,j}, f_i, f_j, f_0 \in \mathbb{F}[\mathbf{x} \setminus \{x_i, x_j\}]$. Then $\Delta_{i,j} = f_{i,j} \cdot f_0 - f_i \cdot f_j$.*

Using this, we can easily prove the following observation, which would be used in Section 3.

► **Observation 15.** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a multilinear polynomial and $i \neq j \in [n]$ such that $\frac{\partial^2 f}{\partial x_i \partial x_j} \equiv 0$. Then, $\Delta_{i,j}f = -\frac{\partial f}{\partial x_i} \cdot \frac{\partial f}{\partial x_j}$.*

► **Fact 16 (Lemma 4.6 of [51]).** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a non-constant multilinear polynomial and $i \neq j \in [n]$. There exist $g, h \in \mathbb{F}[\mathbf{x}]$ where $i \notin \text{var}(h)$ and $j \notin \text{var}(g)$ such that $f = g \cdot h$ if and only if $\Delta_{i,j}f \equiv 0$.*

The fact above implies the following result.

► **Corollary 17.** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a multilinear polynomial, where $n \geq 2$. Then, f is reducible if and only if there exist $i, j \in \text{var}(f)$ such that $\Delta_{i,j}f \equiv 0$.*

The following property of a commutator immediately follows from Definition 13.

► **Observation 18.** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a multilinear polynomial, $\mathbf{a} \in \mathbb{F}^n, i \neq j \in \text{var}(f)$, and $I \subseteq \text{var}(f) \setminus \{i, j\}$. Then, $\Delta_{i,j}(f|_{\mathbf{x}_I=\mathbf{a}_I}) = (\Delta_{i,j}f)|_{\mathbf{x}_I=\mathbf{a}_I}$.*

The following useful claim relates commutators and irreducibility preserving assignments, which would play an important role in Section 3.1.4.

▷ **Claim 19 (Commutators and irreducibility preserving assignments).** *Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a multilinear polynomial and $\mathbf{a} \in \mathbb{F}^n$ s.t. $f(\mathbf{a}) \neq 0$. Suppose that for every $i \neq j \in \text{var}(f)$: $(\Delta_{i,j}f)(\mathbf{a}) \neq 0$. Then, f is \mathbf{a} -irreducible.*

Proof. Suppose f is not \mathbf{a} -irreducible. Then, either $f(\mathbf{a}) = 0$ or there exists a proper subset $I \subsetneq \text{var}(f)$ such that $f|_{\mathbf{x}_I=\mathbf{a}_I}$ is reducible. In the former case, we immediately get a contradiction. Now, suppose the latter holds. Then, there exist non-constant multilinear polynomials $g, h \in \mathbb{F}[\mathbf{x}]$ such that $f|_{\mathbf{x}_I=\mathbf{a}_I} = g \cdot h$. Let $i \in \text{var}(g)$ and $j \in \text{var}(h)$. As $f|_{\mathbf{x}_I=\mathbf{a}_I}$ is multilinear, g and h are variable disjoint. Then, it follows from Corollary 17 that $\Delta_{i,j}(f|_{\mathbf{x}_I=\mathbf{a}_I}) \equiv 0$, which implies $(\Delta_{i,j}(f|_{\mathbf{x}_I=\mathbf{a}_I}))(\mathbf{a}) = 0$. Observation 18 implies that $(\Delta_{i,j}(f|_{\mathbf{x}_I=\mathbf{a}_I}))(\mathbf{a}) = (\Delta_{i,j}f)(\mathbf{a})$. Since $(\Delta_{i,j}(f|_{\mathbf{x}_I=\mathbf{a}_I}))(\mathbf{a}) = 0$, we get $(\Delta_{i,j}f)(\mathbf{a}) = 0$, which is a contradiction. Hence, f is \mathbf{a} -irreducible. ◁

2.4 The Generator $G_{n,k}$ of [53]

Due to limitations of space, see [8] for formal definitions of a hitting set and a generator. Here, we discuss the specific generator $G_{n,k}$, which was defined in [53] for the class of ROFs. It has been a crucial ingredient in PIT algorithms of various other interesting classes also [31, 20, 5, 41]. We will also be using this generator in our results. We borrow the definition and properties of this generator as presented in [5, 55].

► **Definition 20.** Let $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ be n distinct elements and for $i \in [n]$, let $L_i(x) \triangleq \prod_{j \in [n] \setminus \{i\}} \frac{x - \alpha_j}{\alpha_i - \alpha_j}$ denote the corresponding Lagrange interpolant. For every $k \in [n]$, let $G_{n,k} : \mathbb{F}^{2k} \rightarrow \mathbb{F}^n$ be defined as

$$G_{n,k}(y_1, \dots, y_k, z_1, \dots, z_k) \triangleq \left(\sum_{j=1}^k L_1(y_j)z_j, \sum_{j=1}^k L_2(y_j)z_j, \dots, \sum_{j=1}^k L_n(y_j)z_j \right)$$

Let $(G_{n,k})_i$ denote the i^{th} component of $G_{n,k}$ and we call α_i as the Lagrange constant associated with this i^{th} component. We can also define G_k to be the class of generators $\{G_{n,k}\}_{n \in \mathbb{N}}$ for all output lengths.

For two generators $\mathcal{G}_1, \mathcal{G}_2$ with the same output length, we define their sum $\mathcal{G}_1 + \mathcal{G}_2$ as their component-wise addition, where the seed variables of both generators are implicitly relabelled so as to be disjoint. With this terminology, we can note various useful properties of the generator $G_{n,k}$ from its definition.

► **Fact 21** ([53, 31, 55]). Let k, k' be positive integers.

1. $G_{n,k}(\mathbf{y}, \mathbf{0}) \equiv \mathbf{0}$.
2. $G_{n,k}(y_1, \dots, y_k, z_1, \dots, z_k)|_{y_k = \alpha_i} = G_{n,k-1}(y_1, \dots, y_{k-1}, z_1, \dots, z_{k-1}) + z_k \cdot \mathbf{e}_i$, where \mathbf{e} is the 0-1 vector with a single 1 in coordinate i and α_i the i^{th} Lagrange constant and $G_{n,0} \triangleq \mathbf{0}$.
3. $G_{n,k}(y_1, \dots, y_k, z_1, \dots, z_k) + G_{n,k'}(y_{k+1}, \dots, y_{k+k'}, z_{k+1}, \dots, z_{k+k'}) = G_{n,k+k'}(y_1, \dots, y_{k+k'}, z_1, \dots, z_{k+k'})$.
4. For every $\mathbf{b} \in \overline{\mathbb{F}}^n$ with at most k non-zero components, $\mathbf{b} \in \text{Im}(G_{n,k})$.

It follows that $G_{n,k}$ hits any polynomial containing a low-support monomial.

► **Fact 22** ([53, 20]). Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a polynomial that contains a non-zero monomial of support-size at most k , for some $k \in \mathbb{N}$. Then $f(G_{n,k}) \neq 0$.

The next property follows from Definition 20 and Fact 21 and states that $G_{n,k}$ forms a chain.

► **Observation 23.** Let $f \in \mathbb{F}[x_1, \dots, x_n]$ be a non-zero polynomial and $k \in \mathbb{N}$ such that $f(G_{n,k}) \neq 0$. Then, for every $\ell \geq k$, $f(G_{n,\ell}) \neq 0$.

Let \mathcal{C} be a circuit class over a field \mathbb{F} . Then we define the class,

$$\text{Res}(\mathcal{C}) \triangleq \{ \text{Res}_{x_i}(A, B) \mid A, B \in \mathcal{C} \text{ are irreducible and } i \in \text{var}(A) \cup \text{var}(B) \}.$$

We note that $\mathcal{C} \subseteq \text{Res}(\mathcal{C})$ as for any polynomial $f \in \mathcal{C}$, we can write f as $f = \text{Res}_y(P, Q)$, where $P \triangleq (f+1) \cdot y + 1$ and $Q \triangleq y + 1$ and $y \notin \text{var}(P)$.⁵ The following fact is implicit in [55] and [9]. See a proof in [8].

► **Fact 24** (Generator for $\sum^{[2]} \prod \mathcal{C}$). Let \mathcal{C} be a class of arithmetic circuits over a field \mathbb{F} and \mathcal{G} be a generator for the class $\text{Res}(\mathcal{C})$. Then, $H = \mathcal{G} + G_1$ is a generator for $\sum^{[2]} \prod \mathcal{C}$.

⁵ We are assuming that both the polynomials $(f+1) \cdot y + 1$ and $y + 1$ are also in \mathcal{C} , which is true for all natural classes of polynomials.

2.5 Hardness of Representation

► **Definition 25.** Let $k, n, m \in \mathbb{N}$ and let $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be polynomials. Define $A \triangleq A_1 + \dots + A_k$. We say that the set $\{A_1, \dots, A_k\}$ is m -hard, if and only if either $A \equiv 0$ or for every set $J \subseteq [n]$ of size $|J| = m$, the monomial $\prod_{j \in J} x_j$ does not divide A .

► **Remark 26.** Defining the identically zero polynomial m -hard may seem unnatural or counter-intuitive. However, it is required for technical reasons. In addition, it follows from the definition that if $n < m$ then any set $\{A_1, \dots, A_k\}$ is m -hard.

The following fact has been used in many works like [53, 5] etc. See a proof in [8].

► **Fact 27** (Hardness of representation implies PIT). Let $m, n, k \in \mathbb{N}$ and $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ such that $A \triangleq A_1 + \dots + A_k \neq 0$. Suppose further that for every subset $I \subseteq [n]$, the set of restricted polynomials $\{A_1|_{\mathbf{x}_I=\mathbf{0}_I}, \dots, A_k|_{\mathbf{x}_I=\mathbf{0}_I}\}$ is m -hard. Then A contains a non-zero monomial of support-size at most $(m-1)$ and in particular $A(G_{n,m-1}) \neq 0$. Here $G_{n,m-1}$ is the generator given in Definition 20.

3 ROFs and Multilinear Bounded-Read Arithmetic Formulae

3.1 ROFs and ROPs

► **Definition 28** (Read-once formulas, [53]). Let \mathbb{F} be a field and $\mathbf{x} = \{x_1, \dots, x_n\}$. A read-once formula (in short, ROF) Φ over \mathbb{F} in \mathbf{x} -variables is a binary tree whose leaves are labelled with variables in \mathbf{x} and non-leaf nodes are labelled with $+$ and \times . Every variable in \mathbf{x} labels at most one leaf of Φ and every node of Φ is associated with a pair $(\alpha, \beta) \in \mathbb{F}^2$. The computation in Φ proceeds as follows: A leaf node of Φ labelled with $x \in \mathbf{x}$ and (α, β) computes $\alpha x + \beta$. A node v labelled with $\circ \in \{+, \times\}$ and (α, β) , and having children v_1 and v_2 computes $\alpha(\Phi_{v_1} \circ \Phi_{v_2}) + \beta$, where Φ_{v_i} is the sub-formula of Φ rooted at v_i .

We say that a polynomial $A \in \mathbb{F}[\mathbf{x}]$ is a *read-once polynomial* (in short, ROP) if it is computed by an ROF. Note that every ROP is multilinear.

3.1.1 Some Useful Properties of ROFs and ROPs

The following fact shows that the class of read-once formulas is closed under factorization and partial derivatives.

► **Fact 29** (Lemmas 3.6 and 3.12 of [53]). Let $A \in \mathbb{F}[\mathbf{x}]$ be an ROP, $i \in [n]$, $I \subseteq [n]$, and $\mathbf{a} \in \mathbb{F}^n$. Then, $A|_{\mathbf{x}_I=\mathbf{a}_I}$, $\frac{\partial A}{\partial x_i}$, and factors of A are ROPs.

Below fact shows that we can make ROPs $\mathbf{0}$ -justified by shifting.

► **Fact 30** ([53, 41]). Let $n, k \in \mathbb{N}$, \mathbb{F} be a field, and $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be ROPs. Then, we there exists an $\mathbf{a} \in \text{Im}(G_{n,1})$ in such that for every $t \in [k]$, $A_t(\mathbf{x} + \mathbf{a})$ is $\mathbf{0}$ -justified.

The fact below follows from Theorem 3.10 of [52].

► **Fact 31.** A partial derivative of a $\mathbf{0}$ -justified ROP is also $\mathbf{0}$ -justified.

The next observation follows from Definition 28.

► **Observation 32.** Let $A, B \in \mathbb{F}[\mathbf{x}]$ be two variable disjoint ROPs. Then, $A \cdot B$ is an ROP.

3.1.2 Commutator of an ROP

► **Fact 33** (Lemma 3.14 of [52]). *Let $A \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be an ROP and $i \neq j \in [n]$ such that $\frac{\partial^2 A}{\partial x_i \partial x_j} \neq 0$. Then, there exist variable disjoint ROPs $B(\mathbf{x}), R(\mathbf{x}, y)$ such that $A = R(\mathbf{x}, B(\mathbf{x}))$ and $\Delta_{i,j} A = R(\mathbf{x}, 0) \cdot \frac{\partial^2 A}{\partial x_i \partial x_j}$.*

Facts 29, Observation 15, and Fact 33 imply the following useful result.

► **Corollary 34** (Structure of a commutator of an ROP). *Let $A \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be an ROP and $i \neq j \in [n]$. Then, $\Delta_{i,j} A$ is a product of ROPs in $\mathbb{F}[x_1, x_2, \dots, x_n]$.*

3.1.3 The Hardness of Representation Theorem for Sum of ROPs

► **Fact 35** (Hardness of representation for sum of k $\mathbf{0}$ -justified ROPs, Theorem 6.1 of [53]). *Let $n, k \in \mathbb{N}$ and $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be $\mathbf{0}$ -justified ROPs. Suppose $n \geq 3k$. Then, for every collection of sets $J_1, \dots, J_k \subseteq [n]$ and every collection of field elements $\alpha_1, \dots, \alpha_k \in \mathbb{F}$, the set $\{\alpha_1 \cdot A_1|_{\mathbf{x}_{J_1}=\mathbf{0}_{J_1}}, \dots, \alpha_k \cdot A_k|_{\mathbf{x}_{J_k}=\mathbf{0}_{J_k}}\}$ is $3k$ -hard.*

When $k = 2$, we can, in fact, show that the polynomials are 3-hard rather than 6-hard. See a formal proof in [8].

► **Fact 36** (Hardness of representation for sum of two $\mathbf{0}$ -justified ROPs). *Let \mathbb{F} be a field and $A, B \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be two $\mathbf{0}$ -justified ROPs. Then the set $\{A, B\}$ is 3-hard.*

Using this fact, we give the following useful result used in Section A.

▷ **Claim 37.** Let $n \geq 3$ be a natural number and $A, B \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be two $\mathbf{0}$ -justified ROPs. Suppose there exists a $J \subseteq [n], |J| = 3$ such that for every $j \in J, A|_{x_j=0} = \alpha_j \cdot B|_{x_j=0}$ for some $\alpha_j \in \mathbb{F}$. Then, $A \sim B$.

Proof. Let $j, k \in J$ be distinct. As A, B are $\mathbf{0}$ -justified, it follows from Definition 2 that

$$A|_{x_j=0, x_k=0} = \alpha_j \cdot B|_{x_j=0, x_k=0} \neq 0, \quad A|_{x_j=0, x_k=0} = \alpha_k \cdot B|_{x_j=0, x_k=0} \neq 0.$$

These two equations immediately imply that there exists a non-zero $\alpha \in \mathbb{F}$ such that for every $j \in J, \alpha_j = \alpha$. Hence, for every $j \in J, A|_{x_j=0} = \alpha \cdot B|_{x_j=0}$. Now, suppose $A - \alpha \cdot B \neq 0$. As for every $j \in J, A|_{x_j=0} - \alpha \cdot B|_{x_j=0} \equiv 0$, Fact 8 implies that for every $j \in J, x_j$ divides $A - \alpha \cdot B$. Since B is a $\mathbf{0}$ -justified ROP, $\alpha \cdot B$ is also a $\mathbf{0}$ -justified ROP. Hence, $\prod_{j \in J} x_j$ divides $A - \alpha \cdot B$, which can not happen because of Fact 36. Thus, $A = \alpha \cdot B$. ◁

3.1.4 Obtaining $\mathbf{0}$ -Irreducible ROPs

In this section, we give a procedure to convert an irreducible ROP to a $\mathbf{0}$ -irreducible ROP (Definition 5). In particular, if $A \in \mathbb{F}[\mathbf{x}]$ is an irreducible n -variate ROP then we compute an assignment $\mathbf{a} \in \mathbb{F}^n$ such that A is \mathbf{a} -irreducible. Observation 7 implies that $A(\mathbf{x} + \mathbf{a})$ is a $\mathbf{0}$ -irreducible ROP.

It follows from Claim 19 that if all the commutators of a multilinear polynomial $f \in \mathbb{F}[\mathbf{x}]$ are non-zero and if we can efficiently hit all these commutators, i.e., we can efficiently compute an $\mathbf{a} \in \mathbb{F}^n$ such that for every $i \neq j \in \text{var}(f), (\Delta_{i,j} f)(\mathbf{a}) \neq 0$, then using Observation 7, we transform f to a $\mathbf{0}$ -irreducible polynomial. It follows from Corollary 17 that for every $i \neq j \in \text{var}(f), \Delta_{i,j} f \neq 0$ if and only if f is irreducible. Thus, only irreducible multilinear polynomials are eligible to be transformed into $\mathbf{0}$ -irreducible polynomials. The following fact from [41] would be used in Claim 39.

► **Fact 38** (Theorem 4.2 of [41]). *Let $A \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a non-zero ROP and $G_{n,1}$ be the generator given in Definition 20. Then, $A(G_{n,1}) \neq 0$.*

▷ **Claim 39** (Converting a set of irreducible ROPs to $\mathbf{0}$ -irreducible ROPs). *Let $n, m \in \mathbb{N}$ and $A_1, \dots, A_m \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be irreducible ROPs. Then, there exists an assignment $\mathbf{a} \in \mathbb{F}^n$ in the image of $G_{n,1}$ (see Definition 20) such that $A_\ell(\mathbf{x} + \mathbf{a})$ is a $\mathbf{0}$ -irreducible ROP for every $\ell \in [m]$.*

Proof. As A_1, \dots, A_m are irreducible, Corollary 17 implies that for every $\ell \in [m], i \neq j \in \text{var}(A_\ell), \Delta_{i,j}A_\ell \neq 0$. It follows from Claim 19 that it is sufficient to show that $G_{n,1}$ hits $A_\ell, \Delta_{i,j}A_\ell$ for every $\ell \in [m], i \neq j \in \text{var}(A_\ell)$. Let

$$\Phi(\mathbf{x}) \triangleq \prod_{\ell \in [m]} A_\ell \prod_{i,j \in \text{var}(A_\ell), i \neq j} \Delta_{i,j}A_\ell.$$

As every A_ℓ is irreducible, it is non-zero (recall that every element of \mathbb{F} is reducible). Then, it follows from Corollary 34 that $\Phi(\mathbf{x})$ is a product of non-zero ROPs in $\mathbb{F}[\mathbf{x}]$. Now, Fact 38 and the multiplicative property of generators together imply that $\Phi(G_{n,1}) \neq 0$. Hence, by definition there exists an \mathbf{a} in the image of $G_{n,1}$ such that $\Phi(\mathbf{a}) \neq 0$. Now, Claim 19 and Observation 7 imply that $A_\ell(\mathbf{x} + \mathbf{a})$ is a $\mathbf{0}$ -irreducible ROP for every $\ell \in [m]$. ◁

3.2 Multilinear Bounded-Read Arithmetic Formulae

► **Definition 40** (Read- k formula). *Let \mathbb{F} be a field and $k \in \mathbb{N}$. An arithmetic read- k formula F is a binary tree where every leaf node is labelled either by a variable or an element of \mathbb{F} ; every other node (or internal node) is labelled by either $+$ or \times ; every edge is labelled by an element of \mathbb{F} ; and every variable labels at most k leaves of F . Every leaf node of F computes its label. Suppose v is an internal node of F labelled by $\circ \in \{+, \times\}$ such that v_1, v_2 are the children of v , for every $i \in [2]$, v_i computes $F_{v_i} \in \mathbb{F}[\mathbf{x}]$ and the edge between v and v_i is labelled by $\alpha_i \in \mathbb{F}$. Then, v computes the following polynomial $F_v \triangleq \alpha_1 F_{v_1} \circ \alpha_2 F_{v_2}$. Further, if every node of F computes a multilinear polynomial then F is called a multilinear read- k formula.*

An ROF is a special case of a multilinear bounded-read formula. One of the reasons for studying multilinear bounded-read arithmetic formulae is that developing deep understanding of such formulae might give us good insights about the class of multilinear formulae, which is an important class of arithmetic circuits. Deterministic algorithms for blackbox and whitebox PIT for multilinear bounded-read arithmetic formulae were given in [5]. For the rest of this section, let $k \in \mathbb{N}$ be a fixed constant and \mathcal{C}_k be the class of multilinear read- k formulae over a field \mathbb{F} . The following result would be used in the proof of Theorem 3.

► **Observation 41.** *Let $k \in \mathbb{N}$, and $A, B \in \mathcal{C}_k$ be variable-disjoint. Then, $A \cdot B \in \mathcal{C}_k$.*

The following fact would play a crucial role in the proof of Theorem 3.

► **Fact 42** (Implicit in [5]). *Let $k \in \mathbb{N}$, $m \triangleq (8k \cdot (k+1)^2)^k$, and $\tilde{A}, \tilde{B}, \tilde{R} \in \mathcal{C}_k$ compute n -variate polynomials over a field \mathbb{F} . Let $\ell \triangleq m + 3k \lceil \log n \rceil$. Then, there exists an assignment $\mathbf{a} \in \text{Im}(G_{n,\ell})$ such that the polynomials $A \triangleq \tilde{A}(\mathbf{x} + \mathbf{a}), B \triangleq \tilde{B}(\mathbf{x} + \mathbf{a})$, and $R \triangleq \tilde{R}(\mathbf{x} + \mathbf{a})$ satisfy Properties 1, 2, and 3 given in Theorem 57.*

Due to space constraint, we have shifted the remaining content to the Appendix. Since the appendix is also space bound, we refer to the full version of this paper given in [8] for the proofs of some results used in the appendix. We direct the reader to Sections A and B of the Appendix. These sections contain the proofs of Theorems 1, 2, and 3.

4 Discussion and Future Work

In this work, we give a polynomial-time blackbox PIT algorithm for the class $\sum^{[2]} \amalg \text{ROF}$. We improve upon a result of [39], which gave a whitebox PIT algorithm for the same class. We also took a step forward in solving an open question in [39]. An efficient deterministic PIT algorithm for the class $\sum^{[k]} \wedge \text{ROF}$ was listed as an open problem in [39]. We give a polynomial-time deterministic blackbox PIT algorithm for $\sum^{[3]} \wedge \text{ROF}$. In addition to these two results, we also give a quasi-polynomial-time deterministic blackbox PIT for $\sum^{[3]} \wedge \mathcal{C}_k$, where \mathcal{C}_k is the class of multilinear read- k arithmetic formulae over a field \mathbb{F} . All our results work over any field. The common thread between these three results is the hardness of representation approach (see Section 2.5). Now we list some open questions.

- PIT for $\sum^{[3]} \amalg \text{ROF}$: Our PIT algorithm for $\sum^{[2]} \amalg \text{ROF}$ crucially depends on the fact that the fan-in of the topmost $+$ gate in the circuits of this class is exactly two. In particular, the resultant based approach used in our algorithm only works in the top fan-in equal to two regime. It is not clear how to lift the resultant-based approach to $\sum^{[3]} \amalg \text{ROF}$. Can we come up with some technique that not only yields efficient PIT algorithm for $\sum^{[3]} \amalg \text{ROF}$, but also has the potential to extend to PIT for $\sum^{[k]} \amalg \text{ROF}$, where k is a constant?
- PIT for $\sum^{[k]} \wedge \text{ROF}$: An efficient PIT algorithm for this class would solve an open question given in [39]. Our PIT algorithm for $\sum^{[3]} \wedge \text{ROF}$ is based on a hardness of representation theorem, which we prove for this class. Can we prove the hardness of representation for $\sum^{[k]} \wedge \text{ROF}$? In this direction, we note the following conjecture.

► **Conjecture 43.** *Let $k, n \in \mathbb{N}$, and $A_1, \dots, A_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be $\mathbf{0}$ -justified ROPs. Then, there exists a monotone function $\varphi : \mathbb{N} \rightarrow \mathbb{N}$ such that for any $e_1, \dots, e_k \in \mathbb{N}$, the set $\{A_1^{e_1}, \dots, A_k^{e_k}\}$ is $\varphi(k)$ -hard (see Definition 25).*

We remark that this conjecture is true when every $e_i = 1$. In particular, [53] showed that for any constant k , the set $\{A_1, \dots, A_k\}$ is $3k$ -hard (see Fact 35). In addition, for the special case when the A_i -s are products of linear forms over the reals, it was shown in [53], based on a result of [47], that set $\{A_1^{e_1}, \dots, A_k^{e_k}\}$ is $R_{\mathbb{R}}(k)$ -hard (for arbitrary e_i -s) where $R_{\mathbb{R}}(k)$ is the so-called “Rank Bound over the reals”. Finally, in [33] it was shown that $R_{\mathbb{R}}(k) = k^{O(k)}$ and improved to $R_{\mathbb{R}}(k) = O(k^2)$ in [49].

- PIT for $\sum^{[2]} \amalg \mathcal{C}_k$: The approach used in the proof of Theorem 1 would immediately solve this problem, provided we are able to efficiently compute a common irreducibility preserving assignment (see Definition 5) of a set of multilinear read- k arithmetic formulae. We know that if we could efficiently hit all the commutators of these formulae then such an assignment can be computed efficiently (see Claim 19). In case of ROFs, it turns out that a commutator of an ROF is a product of ROF (see Corollary 34). What can we say about the structure of commutators of multilinear bounded-read arithmetic formulae?

References

- 1 M. Agrawal. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th FSTTCS*, volume 3821 of *LNCS*, pages 92–105, 2005.
- 2 M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160(2):781–793, 2004.
- 3 M. Agrawal, C. Saha, R. Saptharishi, and N. 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.

- 4 M. Agrawal, C. Saha, and N. Saxena. Quasi-polynomial hitting-set for set-depth- δ formulas. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC)*, pages 321–330, 2013.
- 5 M. Anderson, D. van Melkebeek, and I. Volkovich. Derandomizing polynomial identity testing for multilinear constant-read formulae. *Computational Complexity*, 24(4):695–776, 2015.
- 6 D. Angluin, L. Hellerstein, and M. Karpinski. Learning read-once formulas with queries. *J. ACM*, 40(1):185–210, January 1993.
- 7 V. Bhargava, S. Saraf, and I. Volkovich. Linear independence, alternants and applications. In *STOC '23: 55th Annual ACM SIGACT Symposium on Theory of Computing, Orlando, Florida, June 20-23, 2023*. ACM, 2023.
- 8 P. Bisht, N. Gupta, and I. Volkovich. Towards identity testing for sums of products of read-once and multilinear bounded-read formulae. In *Electronic Colloquium on Computational Complexity*, 2023. URL: <https://eccc.weizmann.ac.il/report/2023/109/>.
- 9 P. Bisht and I. Volkovich. On solving sparse polynomial factorization related problems. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India*, volume 250 of *LIPICs*, pages 10:1–10:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICs.FSTTCS.2022.10.
- 10 D. Bshouty and N. H. Bshouty. On interpolating arithmetic read-once formulas with exponentiation. *JCSS*, 56(1):112–124, 1998.
- 11 N. H. Bshouty and R. Cleve. Interpolating arithmetic read-once formulas in parallel. *SIAM J. on Computing*, 27(2):401–413, 1998.
- 12 N. H. Bshouty, T. R. Hancock, and L. Hellerstein. Learning arithmetic read-once formulas. *SIAM J. on Computing*, 24(4):706–735, 1995.
- 13 N. H. Bshouty, T. R. Hancock, and L. Hellerstein. Learning boolean read-once formulas with arbitrary symmetric and constant fan-in gates. *JCSS*, 50:521–542, 1995.
- 14 N.H. Bshouty, T.R. Hancock, and L. Hellerstein. Learning boolean read-once formulas over generalized bases. *J. Comput. Syst. Sci.*, 50(3):521–542, June 1995.
- 15 D. A. Cox, J. Little, and D. O’Shea. *Ideals, varieties, and algorithms – An introduction to computational algebraic geometry and commutative algebra (4. ed.)*. Undergraduate texts in mathematics. Springer, 2015.
- 16 R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978.
- 17 P. Dutta, P. Dwivedi, and N. Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPICs*, pages 11:1–11:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 18 Z. Dvir and A. Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. *SIAM J. on Computing*, 36(5):1404–1434, 2007.
- 19 M. A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In *FOCS*, 2015.
- 20 M. A. Forbes, R. Saptharishi, and A. Shpilka. Pseudorandomness for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC)*, pages 867–875, 2014. Full version at <https://eccc.weizmann.ac.il/report/2013/132>. doi:10.1145/2591796.2591816.
- 21 M. A. Forbes and A. Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *APPROX-RANDOM*, pages 527–542, 2013.
- 22 J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.
- 23 K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer, 1992.
- 24 N. Gupta, C. Saha, and B. Thankey. Equivalence test for read-once arithmetic formulas. In Nikhil Bansal and Viswanath Nagarajan, editors, *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 4205–4272. SIAM, 2023.

- 25 R. Gurjar, A. Korwar, and N. Saxena. Identity testing for constant-width, and commutative, read-once oblivious abps. In *31st Conference on Computational Complexity, CCC*, pages 29:1–29:16, 2016. doi:10.4230/LIPIcs.CCC.2016.29.
- 26 R. Gurjar, A. Korwar, N. Saxena, and N. Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. In *30th Conference on Computational Complexity, CCC*, pages 323–346, 2015. doi:10.4230/LIPIcs.CCC.2015.323.
- 27 T. R. Hancock and L. Hellerstein. Learning read-once formulas over fields and extended bases. In *Proceedings of the 4th Annual Workshop on Computational Learning Theory (COLT)*, pages 326–336, 1991.
- 28 J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC)*, pages 262–272, 1980.
- 29 V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 355–364, 2003.
- 30 M. Karchmer, N. Linial, I. Newman, M. Saks, and A. Wigderson. Combinatorial characterization of read-once formulae. *Discrete Math.*, 114(1–3):275–282, April 1993.
- 31 Z. S. Karnin, P. Mukhopadhyay, A. Shpilka, and I. Volkovich. Deterministic identity testing of depth 4 multilinear circuits with bounded top fan-in. *SIAM J. on Computing*, 42(6):2114–2131, 2013.
- 32 Z. S. Karnin and A. Shpilka. Deterministic black box polynomial identity testing of depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 23rd Annual IEEE Conference on Computational Complexity (CCC)*, pages 280–291, 2008.
- 33 N. Kayal and S. Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 198–207, 2009. Full version at <https://eccc.weizmann.ac.il/report/2009/032>.
- 34 N. Kayal and N. Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.
- 35 A. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–223, 2001.
- 36 M. Kumar and R. Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019.
- 37 N. Limaye, S. Srinivasan, and S. Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 804–814. IEEE, 2021.
- 38 L. Lovasz. On determinants, matchings, and random algorithms. In L. Budach, editor, *Fundamentals of Computing Theory*. Akademie-Verlag, 1979.
- 39 M. Mahajan, B.V.R. Rao, and K. Sreenivasaiiah. Building above read-once polynomials: Identity testing and hardness of representation. *Algorithmica*, 76:890–909, 2016.
- 40 D. Medini and A. Shpilka. Hitting sets and reconstruction for dense orbits in $\text{vp}_{\{e\}}$ and $\Sigma\Pi\Sigma$ circuits. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 19:1–19:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 41 D. Minahan and I. Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *TOCT*, 10(3):10:1–10:11, 2018. doi:10.1145/3196836.
- 42 M. Neunhöffer, 2007. Lecture notes on finite fields - Module MT 5826, Chapter 4, Link - <http://www.math.rwth-aachen.de/homes/Max.Neunhoeffer/Teaching/ff/ffchap4.pdf>.
- 43 S. Peleg and A. Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via edelstein-kelly type theorem for quadratic polynomials. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 259–271. ACM, 2021.

- 44 C. Ramya and B.V.R. Rao. Lower bounds for sum and sum of products of read-once formulas. *ACM Transactions on Computation Theory (TOCT)*, 11(2):1–27, 2019.
- 45 C. Saha and B. Thankey. Hitting sets for orbits of circuit classes and polynomial families. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPICs*, pages 50:1–50:26. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
- 46 S. Saraf and I. Volkovich. Blackbox identity testing for depth-4 multilinear circuits. *Combinatorica*, 38(5):1205–1238, 2018.
- 47 N. Saxena and C. Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM J. Comput.*, 40(1):200–224, 2011.
- 48 N. 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.
- 49 N. Saxena and C. Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33, 2013.
- 50 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.
- 51 A. Shpilka and I. Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *Automata, Languages and Programming, 37th International Colloquium (ICALP)*, pages 408–419, 2010. Full version at <https://eccc.weizmann.ac.il/report/2010/036>.
- 52 A. Shpilka and I. Volkovich. On reconstruction and testing of read-once formulas. *Theory of Computing*, 10:465–514, 2014.
- 53 A. Shpilka and I. Volkovich. Read-once polynomial identity testing. *Computational Complexity*, 24(3):477–532, 2015.
- 54 A. Sinhababu and T. Thierauf. Factorization of polynomials given by arithmetic branching programs. *computational complexity*, 30(2):1–47, 2021.
- 55 I. Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *APPROX-RANDOM*, pages 943–958, 2015.
- 56 I. Volkovich. Characterizing arithmetic read-once formulae. *ACM Transactions on Computation Theory (ToCT)*, 8(1):2, 2016. doi:10.1145/2858783.
- 57 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 216–226, 1979.

A PIT for $\Sigma^{[2]} \amalg$ ROF

This section is devoted to the proof of Theorem 1, which is based on the proof overview given in Section 1.3.1. We first present some results related to the resultant of two co-prime and $\mathbf{0}$ -irreducible ROPs in Section A.1. These results are required for proving Theorem 48. Then, using Theorem 48, we give a proof of Theorem 1 in Section A.2.

A.1 Properties of the Resultant of two $\mathbf{0}$ -Irreducible ROPs

We note some important results related to the resultant of two n -variate, co-prime, and $\mathbf{0}$ -irreducible ROPs A, B . The most important result that we prove is a hardness of representation theorem for $\text{Res}_x(A, B)$ (Lemma 45). A key consequence of this result is that there exists a monomial of support (at most) two in $\text{Res}_x(A, B)$ (Corollary 46). Using this, we show in Lemma 47 that $\text{Res}_x(A, B)(G_{n,3}) \not\equiv 0$, where A, B are any co-prime ROPs (not necessarily $\mathbf{0}$ -irreducible). We start with the following claim. Its proof is given in [8].

▷ **Claim 44.** Let $f, g \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be multilinear polynomials, where f is $\mathbf{0}$ -irreducible and $i \neq j \in [n]$ such that $\text{var}(f) \setminus \{i, j\} \neq \emptyset$. Suppose x_j divide $\text{Res}_{x_i}(f, g)$. Then, $f|_{x_j=0}$ divides $g|_{x_j=0}$.

The following lemma lies at the heart of the proof of Theorem 1. We show that the resultant of two $\mathbf{0}$ -irreducible ROPs is 3-hard (see Definition 25).

► **Lemma 45** (Hardness of representation for the resultant of two $\mathbf{0}$ -irreducible ROPs). *Let $n \geq 3$ be a natural number and $A, B \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be $\mathbf{0}$ -irreducible ROPs. Let $i \in [n]$ be such that $\text{Res}_{x_i}(A, B) \neq 0$. Then $\text{Res}_{x_i}(A, B)$ is 3-hard.*

Proof. Suppose for contradiction that $\text{Res}_{x_i}(A, B)$ is not 3-hard, i.e., $\text{Res}_{x_i}(A, B) \neq 0$ and $\exists J \subseteq [n], |J| = 3$ such that for every $j \in J, x_j$ divides $\text{Res}_{x_i}(A, B)$. We claim that this implies either $J \subseteq \text{var}(A)$ or $J \subseteq \text{var}(B)$. As x_j divides $\text{Res}_{x_i}(A, B)$ for every $j \in J$, we get that $j \in \text{var}(A) \cup \text{var}(B)$. Observe that this implies either $|\text{var}(A) \cap J| \geq 2$ or $|\text{var}(B) \cap J| \geq 2$. Suppose the former is true and $j \neq k \in \text{var}(A) \cap J$. As A is $\mathbf{0}$ -irreducible, it follows from Claim 44 that $A|_{x_j=0}$ divides $B|_{x_j=0}$ and $A|_{x_k=0}$ divides $B|_{x_k=0}$. This implies that $j, k \in \text{var}(B)$. Let $\ell \in J \setminus \{j, k\}$. Since $\ell \in \text{var}(A) \cup \text{var}(B)$, we get that either $J \subseteq \text{var}(A)$ or $J \subseteq \text{var}(B)$.

By using the fact that B is also $\mathbf{0}$ -irreducible and by using a similar argument as above, it follows that $J \subseteq \text{var}(A) \cap \text{var}(B)$. This implies that for every $j \in J$, there exists a non-zero $\alpha_j \in \mathbb{F}$ such that $A|_{x_j=0} = \alpha_j \cdot B|_{x_j=0}$. Since B is $\mathbf{0}$ -irreducible, $|\text{var}(B)| \geq 3$, and $\alpha_j \neq 0$ we get that $\alpha_j \cdot B|_{x_j=0} \neq 0$. Since A and B are $\mathbf{0}$ -irreducible ROPs, Claim 6 implies that these are also $\mathbf{0}$ -justified. Then, it follows from Claim 37 that there exists an $\alpha \neq 0 \in \mathbb{F}$ such that $A = \alpha \cdot B$. But this means that A, B are not co-prime, thus Fact 10 implies that $\text{Res}_{x_i}(A, B) \equiv 0$, which is a contradiction. Hence, $\text{Res}_{x_i}(A, B)$ is 3-hard. ◀

Using Lemma 45, we can now show that $\text{Res}_{x_i}(A, B)$ has a monomial of support-size at most 2 and hence we can hit $\text{Res}_{x_i}(A, B)$ using $G_{n,2}$. Note that $\text{Res}_{x_i}(A, B)$ by definition does not depend on x_i but we can still consider it as a polynomial in $\mathbb{F}[x_1, \dots, x_n]$. Henceforth, we follow this convention for any $(n-1)$ -variate polynomial.

► **Corollary 46.** *Let $A, B \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be $\mathbf{0}$ -irreducible ROPs such that $\text{Res}_{x_i}(A, B) \neq 0$ for some $i \in [n]$. Then, $\text{Res}_{x_i}(A, B)$ contains a monomial of support-size at most 2. In particular, $(\text{Res}_{x_i}(A, B))(G_{n,2}) \neq 0$.*

Proof. Let $R \triangleq \text{Res}_{x_i}(A, B) \neq 0$. Let J be any subset of $\text{var}(R)$. Note that $i \notin J$. Since A, B are $\mathbf{0}$ -irreducible and multilinear, we have that $\deg_{x_i}(P) = \deg_{x_i}(P|_{\mathbf{x}_J=\mathbf{0}_J}) = 1$ for both $P = A, B$. Then by Fact 10,

$$R|_{\mathbf{x}_J=\mathbf{0}_J} = \text{Res}_{x_i}(A, B)|_{\mathbf{x}_J=\mathbf{0}_J} = \text{Res}_{x_i}(A|_{\mathbf{x}_J=\mathbf{0}_J}, B|_{\mathbf{x}_J=\mathbf{0}_J}). \quad (1)$$

Since A, B are $\mathbf{0}$ -irreducible, both $A|_{\mathbf{x}_J=\mathbf{0}_J}, B|_{\mathbf{x}_J=\mathbf{0}_J}$ are also $\mathbf{0}$ -irreducible. Then by Lemma 45 we deduce that $R|_{\mathbf{x}_J=\mathbf{0}_J}$ is also 3-hard for any $J \subseteq \text{var}(R)$. Now, we can use Fact 27 to show that R has a monomial of support-size at most 2 and thus $R(G_{n,2}) \neq 0$. ◀

In Corollary 46 we showed how to hit the resultant of two $\mathbf{0}$ -irreducible ROPs. In the lemma below, we show how to hit resultant of two general ROPs. A proof of this lemma is given in [8]. Recall the definition of the class $\text{Res}(\mathcal{C})$ and Fact 24 from Section 2.

► **Lemma 47.** G_3 is a generator for the class $\text{Res}(\text{ROF})$.

A.2 Proof of Theorem 1: PIT for $\sum^{[2]} \prod$ ROF

In Section 1.3, we outlined why it suffices to hit the resultants of two ROFs in order to hit the class $\sum^{[2]} \prod$ ROF. For a general class \mathcal{C} , it is implicitly shown in previous works like [55, 9] that it suffices to hit the class $\text{Res}(\mathcal{C})$, which we formally stated in Fact 24. In Lemma 47, we have shown that G_3 is a generator for $\text{Res}(\text{ROF})$. As a consequence, we get that $G_{n,4} = G_{n,3} + G_{n,1}$ is a generator for any n -variate polynomial in the class $\sum^{[2]} \prod$ ROF.

► **Theorem 48** (G_4 hits $\sum^{[2]} \prod$ ROF). *Let $f \in \sum^{[2]} \prod$ ROF be an n -variate polynomial and $G_{n,4}$ be the generator given in Definition 20. Then, $f \equiv 0$ if and only if $f(G_{n,4}) \equiv 0$.*

Proof. In Lemma 47, we showed that G_3 is a generator for the class $\text{Res}(\text{ROF})$. Then by Fact 24, $G_4 = G_3 + G_1$ is a generator for the class $\sum^{[2]} \prod$ ROF, that is, given an n -variate polynomial $f \in \sum^{[2]} \prod$ ROF, $f \equiv 0$ if and only if $f(G_{n,4}) \equiv 0$. ◀

Proof of Theorem 1. We are given an n -variate polynomial $f \in \sum^{[2]} \prod$ ROF such that $\deg(f) \leq d$. Then, Theorem 48 implies that $f(G_{n,4}) \equiv 0$ if and only if $f \equiv 0$. Since $f(G_{n,4})$ is an eight-variate polynomial and has degree at most $n \cdot d$, it is not difficult to show that the zeroness of $f(G_{n,4})$ can be tested in $\text{poly}(n, d)$ time. ◀

B PIT for $\sum^{[3]} \wedge \mathcal{C}$

This section is devoted to the proofs of Theorems 2 and 3. Here, we prove a more general result which subsumes these two theorems (see Theorem 58). Its proof goes via a hardness of representation result given in Theorem 57. We first discuss some useful definitions and facts.

Let \mathcal{I}_ℓ be the ideal in $\mathbb{F}[x_1, x_2, \dots, x_n]$ generated by $\langle x_1 x_2 \cdots x_\ell \rangle$ for some $\ell \leq n$.

► **Definition 49** (polynomial-hat). *For any polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ and monomial ideal \mathcal{I}_ℓ , we can write f as $f = \tilde{f} + \hat{f}$, where $\tilde{f} \in \mathcal{I}_\ell$ and $\hat{f} = f \pmod{\mathcal{I}_\ell}$ is the unique polynomial obtained from f after going modulo \mathcal{I}_ℓ .*

► **Remark 50.** Note that the polynomial \hat{f} may not be unique for general ideals but here for the monomial ideal \mathcal{I}_ℓ , we define it uniquely by removing all the monomials in f which are divisible by $x_1 \cdots x_\ell$.

The fact below gives a necessary and sufficient condition on the existence of an r -th primitive root of unity in the algebraic closure of a field. See Theorem 8.2 of [42] for a proof.

► **Fact 51.** *Let $\overline{\mathbb{F}}$ be the algebraic closure of a field \mathbb{F} and $r \in \mathbb{N}$. Then $\overline{\mathbb{F}}$ contains an r -th primitive root of unity if and only if $r \nmid \text{char}(\mathbb{F})$.*

► **Observation 52.** *Let $e \in \mathbb{N}$, \mathbb{F} be a field containing an e -th primitive root of unity ω , and x, y be two variables. Then, $x^e - y^e = \prod_{\ell \in [e]} (x - \omega^\ell y)$.*

B.1 Some Useful Results

In this section, we give a set of results required for the proofs of Theorems 2 and 3. The following result generalizes Claim 37. A proof of Claim 53 is given in [8].

▷ **Claim 53.** Let $n, m \in \mathbb{N}$, $n \geq m \geq 2$ and $A, B \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be two $\mathbf{0}$ -justified polynomials such that for every $\beta \in \mathbb{F}$, the set $\{A, \beta \cdot B\}$ is m -hard. Suppose there exists a set $J \subseteq [n]$, $|J| = m$ such that for every $j \in J$, there exists an $\alpha_j \in \mathbb{F}$ satisfying $A|_{x_j=0} = \alpha_j \cdot B|_{x_j=0}$. Then, $A \sim B$.

The proofs of the Claims 54, 56, and Lemma 55 are given in the full version [8].

▷ **Claim 54.** Let \mathbb{F} be a field, $n, e, d \in \mathbb{N}$, such that $2 \leq d \leq e$ and $\text{char}(\mathbb{F})$ does not divide e . Let $f, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be multilinear polynomials such that h is non-constant. Suppose $f^e - g^e = h^d$. Then, we get the following.

1. If $d \geq 2$ then $d = e$ and $f \sim g \sim h$.
2. If $d = 1$ then $e = 2$.

Let $n, r \in \mathbb{N}, r \leq n$, \mathcal{P}_r be the monomial $x_1 \dots x_r$, and $\mathcal{I}_r \triangleq \langle \mathcal{P}_r \rangle$ be the monomial ideal in $\mathbb{F}[x_1, x_2, \dots, x_n]$. For the lemma below, recall Definition 49. Here we consider a polynomial f w.r.t. ideal \mathcal{I}_r and write $f = \tilde{f} + \hat{f}$, where $\hat{f} = f \pmod{\mathcal{I}_r}$.

► **Lemma 55.** Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be a non-constant $\mathbf{0}$ -justified multilinear polynomial such that $f = g \cdot h + v \cdot \mathcal{P}_r$, where $3 \leq r \leq n$ and $v, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$ are arbitrary polynomials (possibly non-multilinear). Then \hat{g} and \hat{h} are variable disjoint.

We shall now use Lemma 55 to prove the following claim that will be used inside the proof of Theorem 57. In this claim, for a polynomial f , we work with the ideal \mathcal{I}_{m+1} and express f as $f = \tilde{f} + \hat{f}$, where $\hat{f} = f \pmod{\mathcal{I}_{m+1}}$.

▷ **Claim 56.** Let $n \geq 3$ be a natural number and let $A, B, R \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be multilinear polynomials that satisfy properties 1, 2 in Theorem 57. Let $H_1 \triangleq A - B$, $H_2 \triangleq A + B$ and $F = H_1 \cdot H_2 - R = v \cdot \mathcal{P}_{m+1}$. For each $i \in \{1, 2\}$, let $J_i = \text{var}(\hat{H}_i)$ and $I_i = [n] \setminus J_i$. Then $J_1 \cap J_2 = \emptyset$, $H_1 \sim R|_{\mathbf{x}_{I_1}=\mathbf{0}_{I_1}}$ and $H_2 \sim R|_{\mathbf{x}_{I_2}=\mathbf{0}_{I_2}}$.

B.2 The Hardness of Representation Theorem and PIT

The theorem below is the main technical result of this section.

► **Theorem 57** (Hardness of representation for $A^{e_1} - B^{e_2} - R^{e_3}$). Let $m \geq 2$. Suppose $A, B, R \in \mathbb{F}[x_1, x_2, \dots, x_n]$ are multilinear polynomials which satisfy the following properties:

1. A, B , and R are $\mathbf{0}$ -justified.
2. For every $J_1, J_2, J_3 \subseteq [n]$ and $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{F}$, the set of polynomials $\{\alpha_1 \cdot A|_{\mathbf{x}_{J_1}=\mathbf{0}_{J_1}}, \alpha_2 \cdot B|_{\mathbf{x}_{J_2}=\mathbf{0}_{J_2}}, \alpha_3 \cdot R|_{\mathbf{x}_{J_3}=\mathbf{0}_{J_3}}\}$ is m -hard.
3. For any disjoint sets $J_1, J_2 \subseteq [n]$ and for every $\alpha, \beta \in \mathbb{F}$, the set of polynomials $\{\alpha \cdot R|_{\mathbf{x}_{I_1}=\mathbf{0}_{I_1}} \cdot R|_{\mathbf{x}_{I_2}=\mathbf{0}_{I_2}}, \beta \cdot R\}$ is m -hard, where $I_1 = [n] \setminus J_1$ and $I_2 = [n] \setminus J_2$.

Let $e_1, e_2, e_3 \in \mathbb{N}$ such that $e_1 \geq e_2 \geq e_3$. Then, the set $\{A^{e_1}, -B^{e_2}, -R^{e_3}\}$ is $(m+1)$ -hard.

Proof. Let $F \triangleq A^{e_1} - B^{e_2} - R^{e_3}$. To prove that $\{A^{e_1}, -B^{e_2}, -R^{e_3}\}$ is $(m+1)$ -hard, either we have to show that $F \equiv 0$ or for every subset $J \subseteq [n], |J| = m+1$, the monomial $\prod_{j \in J} x_j$ does not divide F . If $F \equiv 0$, there is nothing to prove. If $n < m+1$ then $F \equiv 0$. Therefore, we can assume without loss of generality that $n \geq m+1$ and $F \not\equiv 0$. Assume for the sake of contradiction that there exists a set $J \subseteq [n], |J| = m+1$ such that $\prod_{j \in J} x_j$ divides F . Without loss of generality, let $J = [m+1]$, which implies that the monomial $\mathcal{P}_{m+1} \triangleq x_1 \dots x_{m+1}$ divides F . In other words, $F \in \mathcal{I}_{m+1}$, where \mathcal{I}_{m+1} is the ideal in $\mathbb{F}[x_1, x_2, \dots, x_n]$ generated by \mathcal{P}_{m+1} . For this proof, we can assume without loss of generality that $\mathbb{F} = \overline{\mathbb{F}}$. This is so because if the monomial \mathcal{P}_{m+1} divides F over the field \mathbb{F} then it also divides it over $\overline{\mathbb{F}}$. As $F \in \mathcal{I}_{m+1}$, there exists a non-zero $v \in \mathbb{F}[x_1, x_2, \dots, x_n]$ such that

$$F = v \cdot \mathcal{P}_{m+1}. \quad (2)$$

If $A, B, R \in \mathbb{F}$ then we immediately get a contradiction. So, we assume without loss of generality that A is non-constant (otherwise, B and R are also constants). Now, we analyze the situation in the following cases.

- **Case 1. $e_1 > e_2$:** Let $j \in [m+1]$ be such that $\text{var}(A) \setminus \{j\} \neq \emptyset$. Since $m+1 \geq 2$, such a j always exists. Then, Equation (2) implies $F|_{x_j=0} = (A|_{x_j=0})^{e_1} - (B|_{x_j=0})^{e_2} - (R|_{x_j=0})^{e_3} \equiv 0$. As A is $\mathbf{0}$ -justified, non-constant, and $\text{var}(A) \setminus \{j\} \neq \emptyset$, we get that $A|_{x_j=0}$ is a non-constant polynomial. Since $A|_{x_j=0}, B|_{x_j=0}, R|_{x_j=0}$ are multilinear, $A|_{x_j=0}$ is non-constant, and $e_1 > e_2$, $F|_{x_j=0} \not\equiv 0$. This means that F is not divisible by x_j , which contradicts the assumption that $F \in \mathcal{I}_{m+1}$.
- **Case 2. $e_1 = e_2 \geq e_3$:** Fix $e = e_1, d = e_3$. Then,

$$F = A^e - B^e - R^d. \tag{3}$$

For the further discussion, we need an e -th primitive root of unity ω in \mathbb{F} . As \mathbb{F} is algebraically closed, Fact 51 tells us that if $p \triangleq \text{char}(\mathbb{F})$ does not divide e then ω is always present in \mathbb{F} . If $p = 0$ then ω exists. Suppose p is a prime number and p divides e . Then, there exists an $e' \in \mathbb{N}$ such that $e = e' \cdot p$. As $p = \text{char}(\mathbb{F})$, Equations (2) and (3) imply

$$(A^{e'} - B^{e'})^p - R^d = v \cdot \mathcal{P}_{m+1}. \tag{4}$$

First, suppose that R is non-constant. Clearly, there exists a $j \in [m+1]$ such that $\text{var}(R) \setminus \{j\} \neq \emptyset$. It follows from Equation (4) that $((A|_{x_j=0})^{e'} - (B|_{x_j=0})^{e'})^p = (R|_{x_j=0})^d$. As $\text{var}(R) \setminus \{j\} \neq \emptyset$, $R|_{x_j=0}$ is a non-constant multilinear polynomial. Then, $p|d$.

Now, suppose that $R \in \mathbb{F}$. As \mathbb{F} is algebraically closed, we know that $\alpha \triangleq R^{\frac{1}{p}}$ is present in \mathbb{F} . Then, $R^d = (\alpha)^{d \cdot p}$. In this case, without loss of generality, we can replace R with α . This is so because observe that Properties 1, 2, and 3 of A, B, R remain intact if R is constant and we replace it with any other constant. This implies that $\{A^e, -B^e, -R^d\}$ is $(m+1)$ -hard if and only if $\{A^e, -B^e, -\alpha^{d \cdot p}\}$ is $(m+1)$ -hard.

Thus, in both the cases discussed above i.e., $R \in \mathbb{F}$ and R is non-constant, there exists a $d' \in \mathbb{N}$ such that $d = d' \cdot p$. Then, again using the fact that $p = \text{char}(\mathbb{F})$, it follows from Equation (2) that $F = (F')^p = v \cdot \mathcal{P}_{m+1}$. As $F \in \mathcal{I}_{m+1}$, observe that $F' \in \mathcal{I}_{m+1}$. Thus, we can work with F' instead of F . This argument allows us to assume without loss of generality that p does not divide e . Hence, by Fact 51, we get that an e -th primitive root of unity ω is present in \mathbb{F} . Then, on substituting $x = A$ and $y = B$ in $x^e - y^e$ in Observation 52, we get the following useful factorization of $A^e - B^e$.

$$A^e - B^e = \prod_{\ell \in [e]} (A - \omega^\ell B). \tag{5}$$

This factorization would be immensely helpful for further analysis. We first assume that $e = 1$. As $d \leq e = 1$, Equation (3) implies that $F = A - B - \beta R$, where $\beta \in \{0, 1\}$. It follows from Property 2 that the set $\{A, -B, -\beta R\}$ is m -hard. Then Definition 25 implies that F can not be divisible by any multilinear monomial having support m . This contradicts our assumption that $F \in \mathcal{I}_{m+1}$. Henceforth, we assume that $e \geq 2$.

- **Sub-case 2.a. R is a field constant:** Suppose $R \equiv 0$. It follows from Equations (3) and (5) that for every $j \in [n]$, there exists an $\ell_j \in [e]$ such that $A|_{x_j=0} = \omega^{\ell_j} B|_{x_j=0}$. As A is $\mathbf{0}$ -justified and $n \geq m+1$, there exists a $J \subseteq [n], |J| = m$ such that for every $j \in J, A|_{x_j=0} \not\equiv 0$. Since $m \geq 2$, Claim 53 implies $A \sim B$. Thus, there exists an $\alpha \in \mathbb{F}$ such that $F = \alpha \cdot A^e$. Observe that $A \in \mathcal{I}_{m+1}$. As $n \geq m+1$, we get from Definition 25 that $\{A\}$ is not $(m+1)$ -hard. On the other hand, as $\{A\}$ is m -hard by assumption (see Property 2), observe that it is also $(m+1)$ -hard. This is a contradiction.

Now, suppose $R \in \mathbb{F} \setminus \{0\}$. Let $j \in [m+1]$ be such that $\text{var}(A) \setminus \{j\} \neq \emptyset$. It follows from Equations (2) and (3) that $(A|_{x_j=0})^e - (B|_{x_j=0})^e = (R|_{x_j=0})^d$. Since $R \in \mathbb{F} \setminus \{0\}$, it is not difficult to show that $A|_{x_j=0}, B|_{x_j=0} \in \mathbb{F}$. But this can not happen as $\text{var}(A) \setminus \{j\} \neq \emptyset$, A is non-constant and $\mathbf{0}$ -justified. Thus, $F|_{x_j=0} \neq 0$, which means that x_j does not divide F and hence F is not in \mathcal{I}_{m+1} . This is a contradiction.

- **Sub-case 2.b. $d \geq 2$:** Let $J \subseteq [m+1]$ such that $|J| = m$ and for every $j \in J$, $\text{var}(R) \setminus \{j\} \neq \emptyset$. Let $j \in J$. Since R is $\mathbf{0}$ -justified and non-constant, $R|_{x_j=0}$ is non-constant. Then, Equations (2) and (3) imply $(A|_{x_j=0})^e - (B|_{x_j=0})^e = (R|_{x_j=0})^d$. It follows from Point 1 of Claim 54 that $d = e$ and for every $j \in J$, $A|_{x_j=0} \sim B|_{x_j=0} \sim R|_{x_j=0}$. Since $|J| = m \geq 2$, it is not difficult to see from Claim 53 that $A \sim B \sim R$. Thus, there exists an $\alpha \in \mathbb{F}$ such that $F = \alpha \cdot A^e$. Since, by assumption, $F \in \mathcal{I}_{m+1}$, observe that $A \in \mathcal{I}_{m+1}$. But this can not happen as A is m -hard, which implies that A can not be divisible by any multilinear monomial of support m . Thus, we get a contradiction.
- **Sub-case 2.c. $d = 1$:** From point 2, we get $e = 2$. Then from Equations (2), (3), and (5), we have $F = (A - B)(A + B) - R = v \cdot \mathcal{P}_{m+1}$. Let $H_1 = A - B$ and $H_2 = A + B$. Then, we can write $R = H_1 \cdot H_2 + v' \cdot \mathcal{P}_{m+1}$, where $v' = -v$. Since $n \geq m + 1 \geq 3$, by Lemma 55, $\widehat{H}_1, \widehat{H}_2$ are variable disjoint. Moreover Claim 56 shows that for disjoint sets $J_1 = \text{var}(\widehat{H}_1), J_2 = \text{var}(\widehat{H}_2)$, we have $H_1 \sim R|_{\mathbf{x}_{I_1}=\mathbf{0}_{I_1}}$ and $H_2 \sim R|_{\mathbf{x}_{I_2}=\mathbf{0}_{I_2}}$, where $I_1 = [n] \setminus J_1$ and $I_2 = [n] \setminus J_2$. By Point 3, we deduce that the set $\{H_1 H_2, -R\}$ is m -hard and hence also $(m+1)$ -hard. This contradicts that $F = H_1 H_2 - R \in \mathcal{I}_{m+1}$. ◀

This result above plays a pivotal role in proving the following theorem. It is not difficult to argue that Theorems 2 and 3 are special details of the following theorem. See [8] for a proof of this theorem and for more details. Recall Definition 20 for this theorem.

► **Theorem 58.** *Let $n \in \mathbb{N}$ and let $\widetilde{A}, \widetilde{B}, \widetilde{R} \in \mathbb{F}[x_1, x_2, \dots, x_n]$ be multilinear polynomials. Suppose $F \triangleq \widetilde{A}^{e_1} + \widetilde{B}^{e_2} + \widetilde{R}^{e_3}$, where $e_1, e_2, e_3 \in \mathbb{N}, e_1 \geq e_2 \geq e_3$. Let m be the parameter mentioned in Theorem 57. Let $H : \mathbb{F}^t \rightarrow \mathbb{F}^n$ be a generator such that there exist an assignment $\mathbf{a} \in \text{Im}(H)$ for which the polynomials $A \triangleq \widetilde{A}(\mathbf{x} + \mathbf{a}), B \triangleq \widetilde{B}(\mathbf{x} + \mathbf{a}),$ and $R \triangleq \widetilde{R}(\mathbf{x} + \mathbf{a})$ satisfy Properties 1, 2, and 3 given in Theorem 57. Then, $F \equiv 0$ if and only if $F(H + G_{n,m}) \equiv 0$.*