Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits

Pranjal Dutta
Chennai Mathematical Institute, India (& CSE, IIT Kanpur)

Prateek Dwivedi
Dept. of Computer Science & Engineering, IIT Kanpur

Nitin Saxena
Dept. of Computer Science & Engineering, IIT Kanpur

Abstract

Polynomial Identity Testing (PIT) is a fundamental computational problem. The famous depth-4 reduction (Agrawal & Vinay, FOCS'08) has made PIT for depth-4 circuits, an enticing pursuit. The largely open special-cases of sum-product-of-sum-of-univariates ($\Sigma^{[k]}\Pi\Sigma\wedge$) and sum-product-of-constant-degree-polynomials ($\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$), for constants k, δ, have been a source of many great ideas in the last two decades. For eg. depth-3 ideas (Dvir & Shpilka, STOC'05; Kayal & Saxena, CCC'06; Saxena & Seshadhri, FOCS'10, STOC'11); depth-4 ideas (Beecken,Mittmann & Saxena, ICALP'11; Saha,Saxena & Saptharishi, Comput.Compl.'13; Forbes, FOCS'15; Kumar & Saraf, CCC'16); geometric Sylvester-Gallai ideas (Kayal & Saraf, FOCS'09; Shpilka, STOC'19; Peleg & Shpilka, CCC'20, STOC'21). We solve two of the basic underlying open problems in this work.

We give the *first* polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$. Further, we give the *first* quasipolynomial time *blackbox* PIT for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$. No subexponential time algorithm was known prior to this work (even if $k = \delta = 3$). A key technical ingredient in all the three algorithms is how the *logarithmic derivative*, and its power-series, modify the top Π-gate to $\wedge$.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Polynomial identity testing, hitting set, depth-4 circuits

Digital Object Identifier 10.4230/LIPIcs.CCC.2021.11

Funding *Pranjal Dutta*: Google Ph. D. Fellowship
Nitin Saxena: DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair

Acknowledgements Pranjal thanks CSE, IIT Kanpur for the hospitality.

1 Introduction: PIT & beyond

Algebraic circuits are natural algebraic analog of boolean circuits, with the logical operations being replaced by $+$ and $\times$ operations over the underlying field. The study of algebraic circuits comprise the large study of algebraic complexity, mainly pioneered (and formalized) by Valiant [87]. A central problem in algebraic complexity is an algorithmic design problem, known as Polynomial Identity Testing (PIT): given an algebraic circuit $\mathcal{C}$ over a field $\mathbb{F}$ and input variables $x_1, \ldots, x_n$, determine whether $\mathcal{C}$ computes the identically zero polynomial. PIT has found numerous applications and connections to other algorithmic problems. Among the examples are algorithms for finding perfect matchings in graphs [59, 62, 24], primality testing [4], polynomial factoring [52, 19], polynomial equivalence [21], reconstruction algorithms [48, 83, 44] and the existence of algebraic natural proofs [16, 53]. Moreover, efficient design of PIT algorithms is intrinsically connected to proving strong lower bounds [39, 1, 42, 23, 29, 17, 20]. Interestingly, PIT also emerges in many fundamental results in complexity theory such as $\mathsf{IP} = \mathsf{PSPACE}$ [82, 60], the PCP theorem [10, 11], and the overarching Geometric Complexity Theory (GCT) program towards $\mathsf{P} \neq \mathsf{NP}$ [64, 63, 32, 41].

36th Computational Complexity Conference (CCC 2021).
Editor: Valentine Kabanets; Article No. 11; pp. 11:1–11:27
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

There are broadly two settings in which the PIT question can be framed. In the *whitebox* setup, we are allowed to look inside the wirings of the circuit, while in the *blackbox* setting we can only evaluate the circuit at some points from the given domain. There is a very simple randomized algorithm for this problem - evaluate the polynomial at a random point from a large enough domain. With very high probability, a nonzero polynomial will have a nonzero evaluation; this is famously known as the Polynomial Identity Lemma [66, 18, 89, 81]. It has been a long standing open question to derandomize this algorithm.

For many years, blackbox identity tests were only known for depth-2 circuits (equivalently sparse polynomials) [13, 49]. In a surprising result, Agrawal and Vinay [7] showed that a complete derandomization of blackbox identity testing for just depth-4 algebraic circuits ($\Sigma\Pi\Sigma\Pi$) already implies a near complete derandomization for the general PIT problem. More recent depth reduction results [50, 36], and the bootstrapping phenomenon [2, 55, 34, 9] show that even PIT for very restricted classes of depth-4 circuits (*even* depth-3) would have very interesting consequences for PIT of general circuits. These results make the identity testing regime for depth-4 circuits, a very meaningful pursuit.

Three PITs in one-shot. Following the same spirit, here we solve three important (and open) PIT questions. We give the *first* deterministic polynomial-time whitebox PIT algorithm for the bounded sum-of-product-of-sum-of-univariates circuits ($\Sigma^{[k]}\Pi\Sigma\wedge$) [71, Open Prob. 2]; polynomials computed by these circuits are of the form $\Sigma_{i\in[k]}\Pi_j\,(g_{ij1}(x_1)+\cdots+g_{ijn}(x_n))$ (Theorem 1). In fact, we also design the first quasipolynomial-time blackbox PIT algorithm for the same model (Theorem 2a). To the best of our knowledge, no subexponential time algorithm was known prior to this work. A similar technique also gives a quasipolynomial-time blackbox PIT algorithm for the bounded top and bottom fanin circuits $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ (where k and δ are constants), see Theorem 2b. These circuits compute polynomials of the form $\Sigma_{i\in[k]}\Pi_j g_{ij}(\boldsymbol{x})$, where $\deg(g_{ij})\le\delta$. Even $\delta=2$ was a challenging open problem [56, Open Prob. 2].

Prior works on the related models. In the last two decades, there has been a surge of results on identity testing for restricted classes of bounded depth algebraic circuits (eg. 'locally' bounded independence, bounded read/occur, bounded variables). There have been numerous results on PIT for depth-3 circuits with bounded top fanin (known as $\Sigma^{[k]}\Pi\Sigma$-circuits). Divir and Shpilka [22] gave the first quasipolynomial-time deterministic whitebox algorithm for $k=O(1)$, using rank based methods, which finally lead Karnin and Shpilka [45] to design algorithm of same complexity in the blackbox setting. Kayal and Saxena [47] gave the first polynomial-time algorithm of the same. Later, a series of works in [78, 79, 80, 5] generalized the model and gave $n^{O(k)}$-time algorithm when the algebraic rank of the product polynomials are bounded.

There has also been some progress on PIT for restricted classes of depth-4 circuits. A quasipolynomial-time blackbox PIT algorithm for *multilinear* $\Sigma^{[k]}\Pi\Sigma\Pi$-circuits was designed in [43], which was further improved to a $n^{O(k^2)}$-time deterministic algorithm in [74]. A quasipolynomial blackbox PIT was given in [12, 56] when algebraic rank of the irreducible factors in each multiplication gate as well as the bottom fanin are bounded. Further interesting restrictions like sum of product of fewer variables, and more structural restrictions have been exploited, see [28, 6, 25, 61, 57]. Some progress has also been made for bounded top-fanin and bottom-fanin depth-4 circuits via incidence geometry [35, 84, 68]. In fact, very recently, [69] gave a polynomial-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$-circuits.

Why were the problems open? As mentioned above, people have studied depth-4 PIT only with extra restrictions, mostly due to the limited applicability of the existing techniques: they were tailor-made for the specific models and do not generalize. Eg. the previous methods handle $\delta = 1$ (i.e. linear polynomials at the bottom) or $k = 2$ (via *factoring*, [71]). While $k = 2$ to 3, or $\delta = 1$ to 2 (i.e. 'linear' to 'quadratic') already demands a qualitatively different approach.

Whitebox $\Sigma^{[k]}\Pi\Sigma\wedge$ model generalizes the famous bounded-top-fanin-depth-3 $\Sigma^{[k]}\Pi\Sigma$ of [47]; but their Chinese Remaindering (CR) method, loses applicability and thus fails to solve even a slightly more general model. The blackbox setting involved similar 'certifying path' ideas [79] which could be thought of as general CR. It comes up with an ideal I such that $f \neq 0 \bmod I$ and finally preserves it under a constant-variate linear map. The preservation gets harder (for both $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$) due to the increased non-linearity of the ideal I generators. Intuitively, larger δ, via ideal-based routes, brings us to the Gröbner basis method (which is doubly-exponential-time in n) [88]. We know that ideals even with 3-generators (analogously $k = 4$) already capture the whole ideal-membership problem [73].

The algebraic-geometric approach to $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ has been explored in [12, 35, 61, 33]. The families which satisfy a certain Sylvester–Gallai configuration (called SG-circuits) is the harder case which is conjectured to have constant transcendence degree [35, Conj. 1]. Non-SG circuits is the case where the nonzeroness-certifying-path question reduces to radical-ideal non-membership questions [30]. This is really a variety question where one could use algebraic-geometry tools to design a poly-time blackbox PIT. In fact, very recently, Guo [33] gave a s^{δ^k}-time PIT by constructing explicit variety evasive subspace families. Unfortunately, this is not the case in the ideal non-membership; this scenario makes it much harder to solve $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$. From this viewpoint, radical-ideal-membership explains well why the intuitive $\Sigma^{[k]}\Pi\Sigma$ methods do not extend to $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$.

Interestingly, Forbes [25] found a quasipolynomial-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ using shifted-partial derivative techniques; but it naively fails when one replaces the $\wedge$-gate by Π (the 'measure' becomes too large). The 'duality trick' [75] completely solves whitebox PIT for $\Sigma\wedge\Sigma\wedge$, by transforming it to a read-once oblivious ABP (ROABP); but it is inapplicable to our models with the top Π-gate (due to large waring rank and ROABP-width). A priori, our models are incomparable to ROABP, and thus, the famous PIT algorithms for ROABP [28, 27, 37] are not expected to help either.

Similarly, a naive application of the 'Jacobian' + 'certifying path' technique [5] fails for our models because it is difficult to come up with a *faithful* map (for constant-variate reduction). Kumar and Saraf [56] crucially used that the computed polynomial has low individual degree (such that [23] can be invoked), while in [57] they exploits the low algebraic rank of the polynomials computed below the top Π-gate. Neither of them hold, in general, for our models. Very recently, Peleg and Shpilka [69] gave a poly-time blackbox PIT for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$, via incidence geometry (eg. Edelstein-Kelly theorem involving 'quadratic' polynomials), by solving [35, Conj. 1] for $k = 3, \delta = 2$. The method seems very strenuous to generalize even to 'cubic' polynomials ($\delta = 3 = k$).

PIT for other models. Blackbox PIT algorithms for many restricted models are known. Egs. ROABP related models [70, 40, 3, 37, 38, 27, 8], log-variate circuits [26, 14], certain non-commutative models [31, 58]. We refer to [85, 76, 64, 77, 54, 72] for detailed surveys on PIT and related topics.

1.1 Our results: An analytic detour to three PITs

Though some attempts have been made to solve PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$, no subexponential time PIT for $k \geq 3$ *even* in the whitebox settings is known, see [71, Open Prob. 2]. Our first result exactly addresses this problem and designs a polynomial-time algorithm (Algorithm 1). The technique (we call it DiDI-paradigm, Sec. 1.2) used is very analytic (& 'non-ideal' based).

Throughout the paper, we will work with $\mathbb{F} = \mathbb{Q}$, though all the results hold for field of large characteristic.

▶ **Theorem 1** (Whitebox $\Sigma\Pi\Sigma\wedge$ PIT)**.** *There is a deterministic, whitebox $s^{O(k\,7^k)}$-time PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$. (See Algorithm 1.)*

▶ Remark. **1.** Case $k \le 2$ can be solved by invoking [71, Thm.5.2]; but $k \ge 3$ was open.
2. Our technique *necessarily* blows up the exponent exponentially in k. In particular, it would be interesting to design a subexponential time algorithm when $k = \Theta(\log s)$.
3. It is not clear if the current technique gives PIT for $\Sigma^{[k]}\Pi\Sigma\wedge^{[2]}$ circuits, i.e. sum of *bi*variate polynomials computed and fed into the top product gate.

Next, we go to the blackbox setting and address two models of interest, namely— $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, where k, δ are constants. The prior best algorithms were exponential-time in s. Our work builds on previous ideas for *unbounded* top fanin— (1) Jacobian [5], (2) the known blackbox PIT for $\Sigma\wedge\Sigma\wedge$ and $\Sigma\wedge\Sigma\Pi^{[\delta]}$ [37, 25] —maneuvering with an analytic approach, *via* power-series, which unexpectedly *reduces* the top Π-gate to a $\wedge$-gate.

▶ **Theorem 2** (Blackbox PIT for depth-4)**.** **(a)** *There is a deterministic $s^{O(k \log\log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$.*
(b) *There is a $s^{O(\delta^2\, k\, \log s)}$-time blackbox PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits of size s, over $\mathbb{F}[\boldsymbol{x}]$.*

▶ Remark. **1.** Thm. 2 has a *better* dependence on k, but *worse* on s, than Thm. 1. Our results are quasipoly-time even up to $k, \delta = \mathsf{poly}(\log s)$.
2. Thm. 2a is better than Thm. 2b, because $\Sigma\wedge\Sigma\wedge$ has a faster algorithm than $\Sigma\wedge\Sigma\Pi^{[\delta]}$.
3. Even for $\Sigma^{[3]}\Pi\Sigma\wedge$ and $\Sigma^{[3]}\Pi\Sigma\Pi^{[3]}$ models, we leave the *poly*-time blackbox question open.

1.2 Proof ideas: A technical synopsis

In this section, we overview the proof of Theorems 1-2. Both the proofs are analytic, i.e. they use *logarithmic derivative*, and its power-series expansion; greatly transforming the respective models. The first proof is inductive, while the second is a *one-shot* proof. We remark that in both the cases, we essentially reduce to the well-known 'wedge' models, that have unbounded top fanin, yet for which PITs are known. This reduction is unforeseeable and quite 'power'ful.

The analytic tool that we use, appears in algebra (and complexity theory) through the *formal power series* ring $\mathsf{R}[[x_1, \ldots, x_n]]$ (in short $\mathsf{R}[[\boldsymbol{x}]]$), see [65, 86, 19]. The advantages of the ring $\mathsf{R}[[\boldsymbol{x}]]$ are many. They usually emerge because of the inverse: $(1 - x_1)^{-1} = \sum_{i \ge 0} x_1^i$, which does not make sense in $\mathsf{R}[x]$, but valid in $\mathsf{R}[[\boldsymbol{x}]]$. Other analytic tools used are inspired from Wronskian (aka linear dependence) [51, Thm.7] [46], jacobian (aka algebraic dependence) [12, 5, 67], and logarithmic derivative operator $\text{dlog}_{z_1}(f) = (\partial_{z_1} f)/f$.

Moreover, we will be working with the division operator (eg. $\mathsf{R}(z_1)$, over a certain ring R). The divisions do not come for 'free'— they require invertibility with respect to z_1 throughout (again landing us in $\mathsf{R}[[z_1]]$, see Lem. 17). We define class $\mathcal{C}/\mathcal{D} := \{f/g \mid f \in \mathcal{C}, \mathcal{D} \ni g \neq 0\}$, for circuit classes $\mathcal{C}, \mathcal{D}$, (similarly $\mathcal{C} \cdot \mathcal{D}$ denotes the class taking respective products).

The DiDI-technique [Idea of Theorem 1]. The proof of Thm. 1 is recursive and uses a novel technique that we introduce in this work, called DiDI (Di= Divide, D=Derive, I=Induct). We illustrate it in $k = 3$, which generalizes to any k.

Before going into the technicalities, we want to convey that $k = 3$ is *perhaps* the first non-trivial case-study. While $k = 1$ is the *simplest* case (follows directly using sparse-PIT hitting set [49]), $k = 2$ invokes a strong *irreducibility* property [71, Thm. 5.2]; and neither of them work for $k \geq 3$.

The case $k = 3$ asks to check whether $T_1 + T_2 + T_3 \stackrel{?}{=} 0$, where $T_i \in \Pi\Sigma\wedge$ of deg $< d$. We apply a homomorphism $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ such that $x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$ where Ψ is another homomorphism. The map $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ is a sparse-PIT map s.t. $\Psi(T_i) \neq 0$ for non-zero T_i, using [49], which ensures that the degree of z_2 is polynomially bounded (Theorem 10). Think of the variable z_1 as a degree-*counter* which also helps later to *derive* (the second 'D' of DiDI). Observe that Φ is a nonzeroness preserving 1-1 map:

$$T_1 + T_2 + T_3 \neq 0 \iff \Phi(T_1) + \Phi(T_2) + \Phi(T_3) \neq 0.$$

Denote $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^d\rangle$. We divide (first 'D' of DiDI), by $\Phi(T_3)$, and derive, wrt z_1, to conclude that $T_1 + T_2 + T_3 = f$ over $\mathbb{F}[\boldsymbol{x}]$ implies

$$\partial_{z_1}\left(\frac{\Phi(T_1)}{\Phi(T_3)}\right) + \partial_{z_1}\left(\frac{\Phi(T_2)}{\Phi(T_3)}\right) = \partial_{z_1}\left(\frac{\Phi(f)}{\Phi(T_3)}\right) \quad \text{over } \mathsf{R}(\boldsymbol{x})\ .$$

Denote $\widetilde{T}_1 := \partial_{z_1}(\Phi(T_1)/\Phi(T_3))$ and $\widetilde{T}_2 := \partial_{z_1}(\Phi(T_2)/\Phi(T_3))$. Moreover, $\partial_{z_1}(\Phi(f)/\Phi(T_3)) = 0$, over $\mathsf{R}(\boldsymbol{x})$, if and only if either (1) $\Phi(f)/\Phi(T_3)$ is z_1-free, in which case it is an element of $\mathbb{F}(z_2)$, this can be easily argued by substituting $z_1 = 0$ in the map Φ; or (2) $\mathrm{val}_{z_1}(\partial_{z_1}(\Phi(f)/\Phi(T_3))) \geq d$, which is a contradiction since it implies $\mathrm{val}_{z_1}(\Phi(f)) \geq d+1$. Here, $\mathrm{val}_{z_1}(\cdot)$ denotes the valuation i.e. the maximum power of z_1 dividing it (which easily extends to fractions via $\mathrm{val}_{z_1}(p/q) := \mathrm{val}_{z_1}(p) - \mathrm{val}_{z_1}(q)$). Whenever we talk about val, think of working over $\mathbb{F}(z_2, \boldsymbol{x})(z_1)$; which is a ring notion that helps us *computationally*, and we track the degree of $\boldsymbol{z}$. This discussion summarizes a crucial fact:

$$T_1 + T_2 + T_3 \neq 0 \iff \widetilde{T}_1 + \widetilde{T}_2 \neq 0 \text{ over } R(\boldsymbol{x}), \quad \text{or} \quad \left.\frac{\Phi(f)}{\Phi(T_3)}\right|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}\ .$$

We remark that the $z_1 = 0$ substitution is a natural condition as the derivation forgets the $(\bmod\ z_1)$-part. At the core, the idea is really 'primal': if a polynomial $g(x) \neq 0$, then either its derivative $g'(x) \neq 0$, or its constant-term $g(0) \neq 0$ (note: $g(0) = g \bmod x$).

Note that, the $z_1 = 0$ substitution part is easy by poly-degree restriction on z_2. If it is already $\neq 0$, we are done, otherwise we need to check $\widetilde{T}_1 + \widetilde{T}_2 \neq 0$. Rewrite $\widetilde{T}_i$ as $\Phi(T_i)/\Phi(T_3) \cdot \mathrm{dlog}_{z_1}(\Phi(T_i)/\Phi(T_3))$, where dlog denotes the logarithmic-derivative, i.e. $\mathrm{dlog}_{z_1}(\cdot) = \partial_{z_1}(\cdot)/(\cdot)$.

Convert top Π to $\wedge$: version 1. The map Ψ ensures that $\Phi(T_3)$ is a unit over R. A calculation shows that the action $\mathrm{dlog}(\Sigma\wedge)$ is in $\Sigma\wedge/\Sigma\wedge \in \Sigma\wedge\Sigma\wedge$, over $\mathsf{R}[\boldsymbol{x}]$ (Claim 4). This crucially uses the inverse identity:

$$\frac{1}{1 - a \cdot z_1} = 1 + a \cdot z_1 + \ldots + a^{d-1} \cdot z_1^{d-1} \quad \text{over } \mathsf{R}[\boldsymbol{x}], \tag{1}$$

for $a \in \mathsf{R}[\boldsymbol{x}]$. Since, dlog is additive over a product (Sec. 2), the action puts $\mathrm{dlog}(\Pi\Sigma\wedge/\Pi\Sigma\wedge)$ in $\sum \mathrm{dlog}(\Sigma\wedge)$, so in $\Sigma\wedge\Sigma\wedge$. Thus, both $\widetilde{T}_1$ and $\widetilde{T}_2$ are of the *bloated* form $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}(\boldsymbol{x})$.

Invertibility. The crucial point is that the $\Pi\Sigma\wedge$-circuits are still *invertible* over $R[\boldsymbol{x}]$ as: dlog newly introduces only $\Sigma\wedge\Sigma\wedge$, while the $\Pi\Sigma\wedge$-parts get multiplied by the $\Pi\Sigma\wedge$ within T_i's, which are invertible by Ψ. Thus, such $(\Pi\Sigma\wedge)|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}$; which will be useful later.

Bloated $k = 2$ model. Is the newly 'reduced' model similar to $k = 2$ base-case? It is a more general expression $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge) + (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge)$. Let $\widetilde{T}_1 + \widetilde{T}_2 =: f_1$, over $\mathsf{R}(\boldsymbol{x})$. We know that $f_1 \neq 0$ (by hypothesis). We again apply 'Divide and Derive' of

DiDI; here we divide with the $\widetilde{T}_i$ where val_{z_1} is *minimal*. Wlog, $\mathrm{val}_{z_1}(\widetilde{T}_2) =: v$, is the minimal valuation. Of course, $0 \le v < d$ (strict because of Ψ). Let us define $\mathsf{R}' := \mathbb{F}(z_2)[z_1]/\langle z_1^{d-v-1}\rangle$. Then, $(\widetilde{T}_1/\widetilde{T}_2) + 1 = f_1/\widetilde{T}_2$ over $\mathsf{R}'(\boldsymbol{x})$. This is well-defined as the division is being done by the minimum valuation (Lemma 17); thus after derivation, the modulus goes from z_1^d to z_1^{d-v-1} which is well-defined over $\mathsf{R}'(\boldsymbol{x})$. However, if we *derive*: $\partial_{z_1}(f_1/\widetilde{T}_2) =: f_2$ may become $= 0$ over $\mathsf{R}'(\boldsymbol{x})$. That could happen if and only if:

1. Either, $f_1/\widetilde{T}_2$ is z_1-free; in that case
$$\left.\frac{f_1}{\widetilde{T}_2}\right|_{z_1=0} = \left.\left(\frac{\widetilde{T}_1}{\widetilde{T}_2} + 1\right)\right|_{z_1=0} \in \mathbb{F}(z_2) \cdot \frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge} + 1.$$
This is easy to test using $\Sigma\wedge\Sigma\wedge$ whitebox PIT (Lemma 18) (we keep track of the circuit-size respectively the degree of z_2 and ensure them polynomially bounded),
2. Or, $\mathrm{val}_{z_1}(f_2) \ge d-v-1 \implies \partial_{z_1}(f_1/\widetilde{T}_2) = z_1^{d-v-1}\cdot p$, for some $p \in \mathsf{R}'(\boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p) \ge 0$; this further implies $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma 17). Thus $\mathrm{val}_{z_1}(f_1/\widetilde{T}_2) \ge d-v \implies f_1 = 0$, over $\mathsf{R}(\boldsymbol{x})$, a contradiction.

Thus, we check the easy condition (1). If the $z_1 = 0$ substitution outputs 0, we need to check whether other monomials of z_1 in f_2 survive. This suffices to conclude $f \ne 0$. Thankfully $f_2 = \partial_{z_1}(\widetilde{T}_1/\widetilde{T}_2)$ is now a $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ circuit over $\mathsf{R}'(\boldsymbol{x})$. This is the same analysis as above that converts top Π to $\wedge$. Except, we may not be able to remove $\Sigma\wedge\Sigma\wedge$ from the denominator; so we work with this fractional bloated model. (Note: the reciprocal may not be in the polynomial ring $\mathsf{R}'[\boldsymbol{x}]$, but only in the ring $\mathsf{R}'(\boldsymbol{x})$.)

Finally, identity testing of $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}'(\boldsymbol{x})$ is *easy*: (1) $\Sigma\wedge\Sigma\wedge$ is closed under coefficient extraction with respect to z_1 (Lemma 13), (2) whitebox identity testing is in P for both $\Pi\Sigma\wedge$ (Theorem 10) and $\Sigma\wedge\Sigma\wedge$ (convert it to an ROABP using [75] and invoke [70], see Lemma 18), (3) the degree of z_1, z_2 respectively circuit-size remain polynomially bounded.

For general induction, our bloated model is $\Sigma^{[k]}(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ [1]. More work shows that it is *closed* under DiDI-technique. This is primarily what makes our polynomial-time algorithm possible. For details, refer to Section 3.1 and Algorithm 1

Jacobian hits again [Idea of Theorem 2]. Suppose we want to test $T_1 + \ldots + T_k \stackrel{?}{=} 0$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ (respec. $\Pi\Sigma\wedge$). We associate a famous polynomial— the Jacobian $J(T_1, \ldots, T_r)$ (see Sec. 2). It captures the algebraic independence of $T_1, \ldots, T_r$ assuming this to be a transcendence basis of the T_i's (see Fact 2). If we could find an r-variate linear map Φ, that keeps $T_1, \ldots, T_r$ algebraically independent, then $\Phi(T_1), \ldots, \Phi(T_r)$ are again algebraically independent and it can be shown that for any C: $C(T_1, \ldots, T_k) = 0 \iff C(\Phi(T_1), \ldots, \Phi(T_k)) = 0$ (Fact 1). Such a map is called 'faithful' [5].

The overall idea is to find an *explicit* homomorphism $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$, and then fix $\boldsymbol{x}$ by a hitting-set H' to get a *composed* map Ψ' s.t. $\mathrm{rk}_{\mathbb{F}(\boldsymbol{x})}\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{z})}\Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$ [here $\mathcal{J}$ is the jacobian matrix and $\boldsymbol{T} = (T_1, \ldots, T_r)$]. Next, *extend* this map to $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, \boldsymbol{y}, t]$ s.t. $x_i \mapsto (\sum_{j=1}^{k} y_j t^{ij}) + \Psi'(x_i)$, which is *faithful*. The construction of the map Ψ' is crucial. We efficiently construct it by reducing $\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T}))$ to $\Sigma\wedge\Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge\Sigma\wedge$) circuits, which have *quasi*poly size hitting sets [25] (respec. Lemma 18).

---

1 This is a special case of $\Sigma^{[k]}\Pi\Sigma\wedge\Sigma\wedge$ circuits; which is really depth-6.

Jacobian works. A priori, Jacobian is a difficult determinant to work with, and so is finding a faithful Φ. However, for the special models (in this paper) we are able to design Φ, mainly because of two reasons— (1) Jacobian being defined via partial derivatives, has a nice 'linearizing effect' on the top Π-gates (that are only $r \le k$ many), (2) Jacobian under a homomorphism Ψ has a nice expression (think of this as a generalized dlog-expression):

$$\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T})) \;=\; \Psi(T_1 \cdots T_r) \cdot \sum_{g_1 \in L(T_1), \ldots, g_r \in L(T_r)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1, \ldots, g_r))}{\Psi(g_1 \ldots g_r)} \;. \qquad \text{(see Eqn. 6)}$$

Here, $L(T_i)$ denotes the multiset of sparse polynomials that constitutes T_i. We show: each $1/\Psi(\cdot)$ has a 'small' $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit (respec. $\Sigma\wedge\Sigma\wedge$). The last point requires *invertibility.* Define, $\Psi : x_i \mapsto z_1 x_i + \Psi_1(x_i)$, where $\Psi_1(\cdot)$ is a sparse-PIT map s.t. $\Psi_1 : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ s.t. $\Psi_1(T_i) \neq 0$. Under the Ψ, T_i is a unit over ring $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^D \rangle$, where D is polynomially bounded. The idea behind the map is similar to that of Thm. 1. Next, we sketch why $\Psi(J_{\boldsymbol{x}_r}(\boldsymbol{T}))$ has a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit (respec. $\Sigma\wedge\Sigma\wedge$) of size $s^{O(k)}$ over $\mathsf{R}[\boldsymbol{x}]$.

Convert top Π to $\wedge$: version 2. The critical point is to show that $1/\Psi(g_1 \cdots g_k)$, over $\mathsf{R}[\boldsymbol{x}]$, where $g_i \in \Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge$) has $s^{O(k)}$ size $\Sigma\wedge\Sigma\Pi^{[\delta]}$ (respec. $\Sigma\wedge\Sigma\wedge$) circuit (see Lem. 9): this again follows from the inverse identity Equation 1. We keep track of the degree of $\boldsymbol{z}$ throughout, which eventually is bounded by $s^{O(k)}$. Thus, the H' can be *efficiently* constructed from the hitting set of the respective models (of quasipolynomial size), see Thm. 24 and 18. The map Φ ultimately provides a hitting set for $T_1 + \ldots + T_k$, as we reduce to a PIT of a polynomial over 'few' (roughly equal to k) variables, yielding a QP-time algorithm.

It is important to note that there was no power series in [5]; this really empowers the jacobian technique as it now manifests new reduced models, for which a hitting-set is known. This technique is also inherently map-based. So, it requires a hitting-set and *fails* to give a *poly*-time whitebox PIT for the respective models. For the detailed proof, see Section 3.2.

2 Preliminaries

Before proving the results, we describe some of the assumptions and notations used throughout the paper. $\boldsymbol{x}$ denotes $(x_1, \ldots, x_n)$. $[n]$ denotes $\{1, \ldots, n\}$.

Logarithmic derivative. Over a ring R and a variable y, the logarithmic derivative $\mathrm{dlog}_y : \mathsf{R}[y] \to \mathsf{R}(y)$ is defined as $\mathrm{dlog}_y(f) := \partial_y f / f$; here ∂_y denotes the partial derivative with respect to variable y. One important property of dlog is that it is additive over a product as

$$\mathrm{dlog}_y(f \cdot g) \;=\; \frac{\partial_y(f \cdot g)}{f \cdot g} \;=\; \frac{(f \cdot \partial_y g \;+\; g \cdot \partial_y f)}{f \cdot g} \;=\; \mathrm{dlog}_y(f) + \mathrm{dlog}_y(g).$$

We refer this effect as *linearization* of product.

Circuit size. Sparsity $\mathrm{sp}(\cdot)$ refers to the number of nonzero monomials. In this paper, it is a parameter of the circuit size. In particular, $\mathrm{size}(g_1 \cdots g_s) = \sum_{i \in [s]} (\mathrm{sp}(g_i) + \deg(g_i))$, for $g_i \in \Sigma\wedge$ (respec. $\Sigma\Pi^{[\delta]}$). In whitebox settings, we also include the *bit-complexity* of the circuit (i.e. bit complexity of the constants used in the wires) in the size parameter. Some of the complexity parameters of a circuit are *depth* (number of layers), *syntactic degree* (the maximum degree polynomial computed by any node), *fanin* (maximum number of inputs to a node).

Hitting set. A set of points $\mathcal{H} \subseteq \mathbb{F}^n$ is called a *hitting-set* for a class $\mathcal{C}$ of n-variate polynomials if for any nonzero polynomial $f \in \mathcal{C}$, there exists a point in $\mathcal{H}$ where f evaluates to a nonzero value. A $T(n)$-time hitting-set would mean that the hitting-set can be generated in time $T(n)$, for input size n.

Valuation. Valuation is a map $\mathrm{val}_y : \mathsf{R}[y] \to \mathbb{Z}_{\geq 0}$, over a ring R, such that $\mathrm{val}_y(\cdot)$ is defined to be the maximum power of y dividing the element. It can be easily extended to fraction field $\mathsf{R}(y)$, by defining $\mathrm{val}_y(p/q) := \mathrm{val}_y(p) - \mathrm{val}_y(q)$; where it can be negative.

Field. We denote the underlying field as $\mathbb{F}$ and assume that it is of characteristic 0. All our results hold for other fields (eg. $\mathbb{Q}_p, \mathbb{F}_p$) of *large* characteristic (see Remarks in Section 3.1-3.2).

Jacobian. The Jacobian of a set of polynomials $\mathbf{f} = \{f_1, \ldots, f_m\}$ in $\mathbb{F}[\boldsymbol{x}]$ is defined to be the matrix $\mathcal{J}_{\boldsymbol{x}}(\mathbf{f}) := \left(\partial_{x_j}(f_i)\right)_{m \times n}$. Let $S \subseteq \boldsymbol{x} = \{x_1, \ldots, x_n\}$ and $|S| = m$. Then, polynomial $J_S(\mathbf{f})$ denotes the minor (i.e. determinant of the submatrix) of $\mathcal{J}_{\boldsymbol{x}}(\mathbf{f})$, formed by the columns corresponding to the variables in S. For its useful properties, see Appendix C.

3 Proof of the main theorems

This section proves Theorems 1-2. The proofs are self contained and we assume for the sake of simplicity that the underlying field $\mathbb{F}$ has characteristic 0. When this is not the case, we discuss the corresponding required characteristic as remarks after the respective proofs.

3.1 Proof of Theorem 1

As seen in Section 1.2, we will induct over the bloated model which naturally generalizes $\Sigma\Pi\Sigma\wedge$ circuits and works ideally under the DiDI-techniques. Formally, we define it below.

▶ **Definition 3.** *We call a circuit $\mathcal{C} \in \mathsf{Gen}(k, s)$, over $\mathsf{R}(\boldsymbol{x})$, for any ring R, with parameter k and size-s, if $\mathcal{C} \in \Sigma^{[k]}(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. It computes $f \in \mathsf{R}(\boldsymbol{x})$, if $f = \sum_{i=1}^{k} T_i$, where*

1. *$T_i =: (U_i/V_i) \cdot (P_i/Q_i)$, for $U_i, V_i \in \Pi\Sigma\wedge$, and $P_i, Q_i \in \Sigma\wedge\Sigma\wedge$,*
2. *$\mathrm{size}(T_i) = \mathrm{size}(U_i) + \mathrm{size}(V_i) + \mathrm{size}(P_i) + \mathrm{size}(Q_i)$, and $\mathrm{size}(f) = \sum_{i \in [k]} \mathrm{size}(T_i)$.*

Eg. Size-s $\Sigma^{[k]}\Pi\Sigma\wedge$-circuit $\in \mathsf{Gen}(k, s)$. We will design a *recursive* algorithm.

Proof of Theorem 1. Begin with $T_{i,0} := T_i$ and $f_0 := f$ where $T_{i,0} \in \Pi\Sigma\wedge$; $\sum_i T_{i,0} = f_0$, and f_0 has size $\leq s$. Assume $\deg(f) < d \leq s$; we keep the parameter d separately, to help optimize the complexity later. In every recursive call we work with $\mathsf{Gen}(\cdot,\cdot)$ circuits. As the input case, define $U_{i,0} := T_{i,0}$ and $V_{i,0} := P_{i,0} := Q_{i,0} := 1$. Further define a 1-1 homomorphism $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ such that $x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$. Here, $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$ is a sparse-PIT map [49] s.t. $\Psi(U_{i,0}) \neq 0$, $\forall i \in [k]$ (Theorem 10). Invertibility implies that $f_0 = 0 \iff \Phi(f_0) = 0$. Further, the degree bound of z_2 on $\Phi(T_{i,0})$ is $\mathsf{poly}(s)$. The algorithm is recursive, and first reduces the identity testing from top-fanin k to $k-1$. Note: $k = 1$ is trivial.

0-th step. **Efficient reduction from k to $k-1$.** By assumption, $\sum_{i=1}^{k} T_{i,0} = f_0$ and

$T_{k,0} \neq 0$. Apply Φ both sides. Then divide and derive:

$$\begin{aligned}
\sum_{i\in[k]} T_{i,0} \;=\; f_0 \;\iff\; & \sum_{i\in[k]} \Phi(T_{i,0}) \;=\; \Phi(f_0) \\
\iff\; & \sum_{i\in[k-1]} \frac{\Phi(T_{i,0})}{\Phi(T_{k,0})} + 1 \;=\; \frac{\Phi(f_0)}{\Phi(T_{k,0})} \\
\implies\; & \sum_{i\in[k-1]} \partial_{z_1}\left(\frac{\Phi(T_{i,0})}{\Phi(T_{k,0})}\right) \;=\; \partial_{z_1}\left(\frac{\Phi(f_0)}{\Phi(T_{k,0})}\right) \\
\iff\; & \sum_{i=1}^{k-1} \frac{\Phi(T_{i,0})}{\Phi(T_{k,0})}\cdot \mathrm{dlog}\left(\frac{\Phi(T_{i,0})}{\Phi(T_{k,0})}\right) \;=\; \partial_{z_1}\left(\frac{\Phi(f_0)}{\Phi(T_{k,0})}\right) . \qquad (2)
\end{aligned}$$

Define the following:

- $\mathsf{R}_1 := \mathbb{F}(z_2)[z_1]/\langle z_1^d\rangle$. Note that, Eqn.(2) holds over $\mathsf{R}_1(\boldsymbol{x})$.
- $\widetilde{T}_{i,1} := \Phi(T_{i,0})/\Phi(T_{k,0})\cdot \mathrm{dlog}(\Phi(T_{i,0})/\Phi(T_{k,0}))$, $\forall\, i\in[k-1]$.
- $f_1 := \partial_{z_1}(\Phi(f_0)/\Phi(T_{k,0}))$, over $\mathsf{R}_1(\boldsymbol{x})$.

Definability of $T_{i,1}$ and f_1. It is easy to see that these are well-defined terms. Here, we emphasize that we do not exactly compute/store $\widetilde{T}_{i,1}$ as a fraction where the degree in z_1 is $< d$; instead it is computed/stored as an element in $\mathbb{F}(z_2)(z_1,\boldsymbol{x})$, where z_1 is a formal variable. Formally, we compute $T_{i,1}\in\mathbb{F}(z_2)(z_1,\boldsymbol{x})$, such that $\widetilde{T}_{i,1} = T_{i,1}$, over $\mathsf{R}_1(\boldsymbol{x})$. We keep track of the degree of z_1 and z_2 in $T_{i,1}$. Thus, $\sum_{i\in[k-1]} T_{i,1} = f_1$, over $\mathsf{R}_1(\boldsymbol{x})$.

The 'iff' condition. Equality in Eqn. (2) over $\mathsf{R}_1(\boldsymbol{x})$ is *one-sided*; however we want a $\iff$ condition to efficiently reduce the identity testing. Note that $f_1\neq 0$ implies $\mathrm{val}_{z_1}(f_1) < d =: d_1$. By assumption, $\Phi(T_{k,0})$ is invertible over $\mathsf{R}_1(\boldsymbol{x})$. Further, $f_1 = 0$, over $\mathsf{R}_1(\boldsymbol{x})$, implies –

1. Either, $\Phi(f_0)/\Phi(T_{k,0})$ is z_1-free. This implies $\Phi(f_0)/\Phi(T_{k,0})\in\mathbb{F}(z_2)(\boldsymbol{x})$, which further implies it is in $\mathbb{F}(z_2)$, because of the map Φ (z_1-free implies $\boldsymbol{x}$-free, by substituting $z_1=0$). Also, note that $f_0, T_{k,0}\neq 0$ implies $\Phi(f_0)/\Phi(T_{k,0})$ is a *nonzero* element in $\mathbb{F}(z_2)$. Thus, it suffices to check whether $\Phi(f_0)|_{z_1=0} = \Psi(f_0)$ is non-zero or not. Further, the degree of z_2 in $\Psi(f_0)$ is polynomially bounded.
2. Or, $\partial_{z_1}(\Phi(f_0)/\Phi(T_{k,0})) = z_1^{d_1}\cdot p$ where $p\in\mathbb{F}(z_2)(z_1,\boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p)\geq 0$. By simple power series expansion, one can conclude that $p\in\mathbb{F}(z_2,\boldsymbol{x})[[z_1]]$ (Lemma 17). Hence, $\Phi(f_0)/\Phi(T_{k,0}) = z_1^{d_1+1}\cdot q$ where $q\in F(z_2,\boldsymbol{x})[[z_1]]$, i.e.
$$\Phi(f_0)/\Phi(T_{k,0}) \in \langle z_1^{d_1+1}\rangle_{\mathbb{F}(z_2,\boldsymbol{x})[[z_1]]} \implies \mathrm{val}_{z_1}(\Phi(f_0)) \geq d+1,$$
a contradiction.

Conversely, it is obvious that $f_0 = 0$ implies $f_1 = 0$. Thus, we have proved the following

$$\sum_{i\in[k]} T_{i,0} \neq 0 \;\text{ over } \mathbb{F}[\boldsymbol{x}] \iff \sum_{i\in[k-1]} T_{i,1}\neq 0 \;\text{ over } \mathsf{R}_1(\boldsymbol{x}), \;\text{ or },\; 0\neq \Phi(f_0)|_{z_1=0}\in\mathbb{F}(z_2)\,.$$

Eventually, we show that $T_{i,1}\in(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}_1(\boldsymbol{x})$, with polynomial blowup in size (Claim 4). So, the above circuit is in $\mathsf{Gen}(k-1,\cdot)$, over $\mathsf{R}_1(\boldsymbol{x})$, which we recurse on to finally give the identity testing. The 1-th step is a bit more tricky:

Induction step. Assume that we are in the j-th step ($j\geq 1$). Our induction hypothesis assumes –

1. $\sum_{i\in[k-j]} T_{i,j} = f_j$, over $\mathsf{R}_j(\boldsymbol{x})$, where $\mathsf{R}_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j}\rangle$, and $T_{i,j} \neq 0$.
2. Here, $T_{i,j} =: (U_{i,j}/V_{i,j}) \cdot (P_{i,j}/Q_{i,j})$, where $U_{i,j}, V_{i,j} \in \Pi\Sigma\wedge$, and $P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma\wedge$, each in $\mathsf{R}_j[\boldsymbol{x}]$. Think of them being computed as $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, with the degrees being tracked. Wlog, assume that $\mathrm{val}_{z_1}(T_{k-j,j})$ is the minimal among all $T_{i,j}$'s.
3. $\mathrm{val}_{z_1}(T_{i,j}) \geq 0, \forall i \in [k-j]$. Moreover, $U_{i,j}|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}$ (similarly $V_{i,j}$).
4. $f \neq 0$, over $\mathbb{F}[\boldsymbol{x}] \iff f_j \neq 0$, over $\mathsf{R}_j(\boldsymbol{x})$, or, $\bigvee_{i=0}^{j-1} ((f_i/T_{k-i,i})|_{z_1=0} \neq 0$, over $\mathbb{F}(z_2)(\boldsymbol{x}))$.

We follow the 0-th step, without applying any further homomorphism. Note that the 'or condition' in the last hypothesis is similar to the $j = 0$ case except that there is no Φ: this is because $\Phi(f_0)|_{z_1=0} \neq 0 \iff \Phi(f_0/T_{k,0})|_{z_1=0} \neq 0$. This condition just separates the derivative from the constant-term (as pointed in Section 1.2).
Let $\mathrm{val}_{z_1}(P_{i,j}/Q_{i,j}) =: v_{i,j}$, for $i \in [k-j]$. Note that

$$\min_i \mathrm{val}_{z_1}(T_{i,j}) = \min_i \mathrm{val}_{z_1}(P_{i,j}/Q_{i,j}) = v_{k-j,j}$$

since $\mathrm{val}_{z_1}(U_{i,j}) = \mathrm{val}_{z_1}(V_{i,j}) = 0$ (else we reorder). We remark that $0 \leq v_{i,j} < d_j$ for all i's in j-th step; upper-bound is strict, since otherwise $T_{i,j} = 0$ over $\mathsf{R}_j(x)$.

Min val computation is easy. Finding this min val is *easy*, as we can compute $\mathrm{val}_{z_1}(P_{i,j})$ and $\mathrm{val}_{z_1}(Q_{i,j})$, $\forall i \in [k-j]$. To compute val, note that $\mathrm{coef}_{z_1^e}(P_{i,j})$ and $\mathrm{coef}_{z_1^e}(Q_{i,j})$ are in $\Sigma\wedge\Sigma\wedge$ as well, over $F(z_2)[\boldsymbol{x}]$ (Lemma 13). We can keep track of z_1 degree and thus interpolate to find the minimum $e < d_j$ such that it is $\neq 0$ (implying it to be the respective val).

Efficient reduction from $k-j$ to $k-j-1$. Similar to the 0-th step, we divide and derive:

$$\begin{aligned}\sum_{i\in[k-j]} T_{i,j} \;=\; f_j \iff & \sum_{i\in[k-j-1]} T_{i,j}/T_{k-j,j} \;+\; 1 \;=\; f_j/T_{k-j,j} \\ \implies & \sum_{i\in[k-j-1]} \partial_{z_1}(T_{i,j}/T_{k-j,j}) \;=\; \partial_{z_1}(f_j/T_{k-j,j}) \\ \iff & \sum_{i=1}^{k-j-1} T_{i,j}/T_{k-j,j} \cdot \mathrm{dlog}(T_{i,j}/T_{k-j,j}) \;=\; \partial_{z_1}(f_j/T_{k-j,j}) \qquad (3)\end{aligned}$$

Define the following:
- $\mathsf{R}_{j+1} := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_{j+1}}\rangle$, where $d_{j+1} := d_j - v_{k-j,j} - 1$.
- $\widetilde{T}_{i,j+1} := T_{i,j}/T_{k-j,j} \cdot \mathrm{dlog}(T_{i,j}/T_{k-j,j})$, $\forall\, i \in [k-j-1]$.
- $f_{j+1} := \partial_{z_1}(f_j/T_{k-j,j})$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$.

Definability of $T_{i,j+1}$ and f_{j+1}. By the minimal valuation assumption, it follows that $\mathrm{val}(f_j) \geq v_{k-j,j}$, and thus $\widetilde{T}_{i,j+1}$ and f_{j+1} are all well-defined over $\mathsf{R}_{j+1}(\boldsymbol{x})$. Note that, Eqn. (3) holds over $\mathsf{R}_{j+1}(\boldsymbol{x})$ as $d_{j+1} < d_j$ (because, whatever identity holds true $\mathrm{mod} z_1^{d_j}$ must hold $\mathrm{mod} z_1^{d_{j+1}}$ as well). Hence, we must have $\sum_{i=1}^{k-j-1} \widetilde{T}_{i,j+1} = f_{j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$ [proving induction hypothesis (1)].

Similarly, we emphasize on the fact that we do not exactly compute $\widetilde{T}_{i,j+1}$ mod $z_1^{d_{j+1}}$; instead it is computed as a fraction in $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, with formal z_1. Formally, we compute/store $T_{i,j+1} \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$, such that $\widetilde{T}_{i,j+1} = T_{i,j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$. We keep track of the degree of z_1 and z_2 in $T_{i,j+1}$. Also, by definition, $\mathrm{val}_{z_1}(T_{i,j+1}) \geq 0$ (as we divide by the min val) [proving induction hypothesis (3), first part]. Of course, we have $\sum_{i\in[k-j-1]} T_{i,j+1} = f_{j+1}$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$.

The 'iff' condition. The above Eqn. (3) pioneers to reduce from $k-j$-summands to $k-j-1$. But we want a $\iff$ condition to efficiently reduce the identity testing. If $f_{j+1} \neq 0$, then $\mathrm{val}_{z_1}(f_{j+1}) < d_{j+1}$. Further, $f_{j+1} = 0$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$ implies–

1. Either, $f_j/T_{k-j,j}$ is z_1-free. This implies it is in $\mathbb{F}(z_2)(\boldsymbol{x})$. Now, if indeed $f_0 \neq 0$, then the computed $T_{i,j}$ as well as f_j must be non-zero over $\mathbb{F}(\boldsymbol{z}_2)(z_1, \boldsymbol{x})$, by induction hypothesis (as they are non-zero over $\mathsf{R}_j(\boldsymbol{x})$). However,
$$\left(\frac{T_{i,j}}{T_{k-j,j}}\right)\Bigg|_{z_1=0} = \left(\frac{U_{i,j}\cdot V_{k-j,j}}{U_{k-j,j}\cdot V_{i,j}}\right)\Bigg|_{z_1=0} \cdot \left(\frac{P_{i,j}\cdot Q_{k-j,j}}{P_{k-j,j}\cdot Q_{i,j}}\right)\Bigg|_{z_1=0} \in \mathbb{F}(z_2)\cdot\left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right).$$
Thus,
$$\frac{f_j}{T_{k-j,j}} \in \sum \mathbb{F}(z_2)\cdot\left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right) \in \left(\frac{\Sigma\wedge\Sigma\wedge}{\Sigma\wedge\Sigma\wedge}\right).$$
Here we crucially use that $\Sigma\wedge\Sigma\wedge$ is closed under multiplication (Lemma 15). We show that the degree of z_2 (in denominator and numerator) in each $T_{i,j}/T_{k,j}$ is poly-bounded. Thus, this identity testing can be done in poly-time (Lemma 18). For, detailed time-complexity and calculations, see Claim 4 and its subsequent paragraph.
2. Or, $\partial_{z_1}(f_j/T_{k-j,j}) = z_1^{d_{j+1}}\cdot p$, where $p \in \mathbb{F}(z_2)(z_1,\boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p) \geq 0$. By a simple power series expansion, one concludes that $p \in \mathbb{F}(z_2,\boldsymbol{x})[[z_1]]$ (Lemma 17). Hence, one concludes that
$$\frac{f_j}{T_{k-j,j}} \in \left\langle z_1^{d_{j+1}+1}\right\rangle_{\mathbb{F}(z_2,\boldsymbol{x})[[z_1]]} \implies \mathrm{val}_{z_1}(f_j) \geq d_j,$$
i.e. $f_j = 0$, over $\mathsf{R}_j(\boldsymbol{x})$.

Conversely, $f_j = 0$, over $\mathsf{R}_j(\boldsymbol{x})$, implies
$$\mathrm{val}_{z_1}(f_j) \geq d_j \implies \mathrm{val}_{z_1}\left(\partial_{z_1}\left(\frac{f_j}{T_{k-j,j}}\right)\right) \geq d_j - v_{k-j,j} - 1 \implies f_{j+1} = 0, \text{ over } \mathsf{R}_{j+1}(\boldsymbol{x}).$$
Thus, we have proved that $\sum_{i\in[k-j]} T_{i,j} \neq 0$ over $\mathsf{R}_j(\boldsymbol{x})$ iff
$$\sum_{i\in[k-j-1]} T_{i,j+1} \neq 0 \text{ over } \mathsf{R}_{j+1}(\boldsymbol{x}) \text{ , or , } 0 \neq \left(\frac{f_j}{T_{k-j,j}}\right)\Bigg|_{z_1=0} \in \mathbb{F}(z_2)(\boldsymbol{x}) \text{ .}$$
Therefore induction hypothesis (4) holds. All we need to show is hypothesis (2) and second part of (3). This part is involved in the size-analysis and dlog-computation, discussed below.

Invertibility of $\Pi\Sigma\wedge$-circuits. Before going into the size analysis, we want to remark that the dlog compuation plays a crucial role here. The action $\mathrm{dlog}(\Sigma\wedge\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$, is of poly-size (Lemma 16). What is the action on $\Pi\Sigma\wedge$? dlog distributes the product additively, so it suffices to work with $\mathrm{dlog}(\Sigma\wedge)$; and we show that $\mathrm{dlog}(\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge$ of poly-size. Assuming these, we simplify
$$\frac{T_{i,j}}{T_{k-j,j}} = \frac{U_{i,j}\cdot V_{k-j,j}}{V_{i,j}\cdot U_{k-j,j}}\cdot\frac{P_{i,j}\cdot Q_{k-j,j}}{Q_{i,j}\cdot P_{k-j,j}},$$
and its dlog. Thus, using Eq. (3), $U_{i,(j+1)}$ grows to $U_{i,j}\cdot V_{k-j,j}$ (and similarly $V_{i,(j+1)}$). This also means: $U_{i,(j+1)}|_{z_1=0} \in \mathbb{F}(z_2)\setminus\{0\}$ (proving hypothesis (3), second part).

Size analysis. We will show that $T_{i,j+1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, over $\mathsf{R}_{j+1}(\boldsymbol{x})$, with only polynomial blowup in size. Let $\mathrm{size}(T_{i,j}) \leq s_j$, for $i \in [k-j]$, and $j \in [k]$. Note that, by assumption, $s_0 \leq s$.

▷ Claim 4 (Final size). $T_{1,k-1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ of size $s^{O(k7^k)}$, over $\mathsf{R}_{k-1}(\boldsymbol{x})$.

Proof. Steps $j = 0$ and $j > 0$ are slightly different because of the Φ. However the main idea of using power-series is the same which eventually shows that $\mathrm{dlog}(\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge$.

We first deal with $j = 0$. Let $A - z_1 \cdot B = \Phi(g) \in \Sigma\wedge$, for some $A \in \mathbb{F}(z_2)$ and $B \in \mathsf{R}_1[\boldsymbol{x}]$. Note that $A \neq 0$ because of the map Ψ. Further, $\mathrm{size}(B) \le O(d \cdot \mathrm{size}(g))$, as a single monomial of the form x^e can produce $d + 1$-many monomials. Over $\mathsf{R}_1(\boldsymbol{x})$,

$$\mathrm{dlog}(\Phi(g)) = -\frac{\partial_{z_1}(B \cdot z_1)}{A(1 - \frac{B}{A} \cdot z_1)} = -\frac{\partial_{z_1}(B \cdot z_1)}{A} \cdot \sum_{i=0}^{d_1-1} \left(\frac{B}{A}\right)^i \cdot z_1^i \ . \tag{4}$$

B^i has a trivial $\wedge\Sigma\wedge$-circuit of size $O(d \cdot \mathrm{size}(g))$. Also, $\partial_{z_1}(B \cdot z_1)$ has a $\Sigma\wedge$-circuit of size at most $O(d\cdot\mathrm{size}(g))$. Using waring identity (Lemma 14), we get that each $\partial_{z_1}(B\cdot z_1)\cdot(B/A)^i\cdot z_1^i$ has size $O(i \cdot d \cdot \mathrm{size}(g))$, over $\mathsf{R}_1(\boldsymbol{x})$. Summing over $i \in [d_1 - 1]$, the overall size is at most $O(d_1^2 \cdot d \cdot \mathrm{size}(g)) = O(d^3 \cdot \mathrm{size}(g))$, as $d_0 = d_1 = d$.

For the j-th step, we emphasize that the degree could be larger than d. Assume that syntactic degree of denominator and numerator of $T_{i,j}$ (each in $\mathbb{F}[\boldsymbol{x}, \boldsymbol{z}]$) are bounded by D_j (it is *not* d_j as seen above; this is to save on the trouble of mod-computation at each step). Of course, $D_0 < d \le s$.

For $j > 0$, the above summation in Equation 4 is over $\mathsf{R}_j(\boldsymbol{x})$. However the degree could be D_j (possibly more than d_j) of the corresponding A and B. Thus, the overall size after the power-series expansion would be $O(D_j^2 \cdot d \cdot \mathrm{size}(g))$.

Using Lemma 16, we can show that $\mathrm{dlog}(P_{i,j}) \in \Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ (similarly for $Q_{i,j}$), of size $O(D_j^2 \cdot s_j)$. Also $\mathrm{dlog}(U_{i,j} \cdot V_{k-j,j}) \in \sum \mathrm{dlog}(\Sigma\wedge)$, i.e. sum of action of dlog on $\Sigma\wedge$ (since dlog linearizes product); and it can be computed by the above formulation. Thus, $\mathrm{dlog}(T_{i,j}/T_{k-j,j})$ is a sum of 4-many $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ of size at most $O(D_j^2\, s_j)$ and 1-many $\Sigma\wedge\Sigma\wedge$ of size $O(D_j^2 d_j s_j)$ (from the above power-series computation) [Note: we summed up the $\Sigma\wedge\Sigma\wedge$-expressions from $\mathrm{dlog}(\Sigma\wedge)$ together]. Additionally the syntactic degree of each denominator and numerator (of the $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$) is $O(D_j)$. We rewrite the 4 expressions (each of $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$) and express it as a single $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ using waring identity (Lemma 15), with the size blowup of $O(D_j^{12}\, s_j^4)$; here the syntatic degree blowsup to $O(D_j)$. Finally we add the remaining $\Sigma\wedge\Sigma\wedge$ circuit (of size $O(D_j^3 s_j)$ and degree $O(dD_j)$) to get $O(s_j^5 D_j^{16} d)$. To bound this, we need to understand the degree bound D_j.

Finally we need to multiply $T_{i,j}/T_{k-j,j} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ where each $\Sigma\wedge\Sigma\wedge$ is a product of two $\Sigma\wedge\Sigma\wedge$ expression of size s_j and syntactic degree D_j; clubbed together owing a blowup of $O(D_j \cdot s_j^2)$. Hence multiplying it with $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ expression obtained from dlog computation above gives size blowup of $s_{j+1} = s^7 \cdot D_j^{O(1)} \cdot d$.

Computing $T_{i,j}/T_{k-j,j}$ increases the syntactic degree 'slowly'; which is much less than the size blowup. As mentioned before, the deg-blowup in dlog-computation is $O(dD_j)$ and in the clearing of four expressions, it is just $O(D_j)$. Thus, $D_{j+1} = O(dD_j) \implies D_j = d^{O(j)}$.

The recursion on the size is $s_{j+1} = s_j^7 \cdot d^{O(j)}$. Using $d \le s$ we deduce, $s_j = (sd)^{O(j\cdot 7^j)}$. In particular, s_{k-1}, size after $k-1$ steps is $s^{O(k\cdot 7^k)}$. This computation quantitatively establishes induction hypothesis (2). ◀

Final time complexity. The above proof actually shows that $T_{1,k-1}$ has a 'bloated' circuit of size $s^{O(k\cdot 7^k)}$ over $\mathsf{R}_{k-1}(\boldsymbol{x})$; and that the degree bound on z_2 and z_1 (over $\mathbb{F}(z_2)[z_1, \boldsymbol{x}]$, keeping denominator and numerator 'in place') is $D_{k-1} = d^{O(k)}$. We note that whitebox PIT for both $\Pi\Sigma\wedge$ and $\Sigma\wedge\Sigma\wedge$ is in poly-time (using Thm. 10 & Lem. 18 respectively), and the proof above is constructive: we calculate $U_{i,j+1}$ (and other terms) from $U_{i,j}$ explicitly. Thus, this part can be done in $s^{O(k7^k)}$ time.

What remains is to test the $z_1 = 0$-part of induction hypothesis (4); it could *short-circuit* the recursion much before $j = k - 1$. As we mentioned before, in this case, we need to do a PIT on $\Sigma\wedge\Sigma\wedge$ only. At the j-th step, when we substitute $z_1 = 0$, the size of each $T_{i,j}$ can be at most s_j (by definition). We need to do PIT on a simpler model: $\sum^{[k-j]} \mathbb{F}(z_2)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. We can clear out and express this as a single $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ expression; with a size blowup of $s_j^{O(k-j)} \leq (sd)^{O(j(k-j)7^j)}$. Further, use the fact that $\max_{j\in[k-1]} j(k-j)7^j = (k-1)7^{k-1}$ (see Lemma 19). The degree bound on z_2 remains as before. Finally, use Lemma 18 for the base-case whitebox PIT. Thus, the final time complexity is $s^{O(k\cdot 7^k)}$.

Here we also remark that in $z_1 = 0$ substitution $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ may be undefined. However, we keep track of z_1 degree of numerator and denominator, which will be polynomially bounded as seen in the discussion above. We can easily interpolate and cancel the z_1 power to make it work.

Bit complexity. It is routine to show that the bit-complexity is really what we claim. Initially, the given circuit has bit-complexity s. The main blowup happens due to the dlog-computation which is a poly-size blowup. We also remark that while using Lemma 15 (using Lemma 14), we *may* need to go to a field extension of at most $s^{O(k)}$ (because of the $\varepsilon(i)$ and correspondingly the constants $\gamma_{\varepsilon(2),\ldots,\varepsilon(k)}$, but they still are $s^{O(k)}$-bits). Also, Theorem 10 and Lemma 18 computations blowup bit-complexity polynomially. This concludes the proof. ◀

▶ Remark.
1. The above method does *not* give whitebox PIT (in poly-time) for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, as we donot know poly-time whitebox PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$. However, the above methods do show that whitebox-PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ polynomially *reduces* to whitebox-PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$.
2. DiDI-technique can be used to give whitebox PIT for the general bloated model $\mathsf{Gen}(k, s)$.
3. The above proof works when the characteristic is $\geq d$. This is because the nonzeroness remains *preserved* after derivation wrt z_1.

3.2 Proof of Theorem 2

Here we prove Theorem 2b only. The proof technique of part (a) has analogous calculations (using bottom $\Sigma\wedge$ instead of $\Sigma\Pi^{[\delta]}$); see Appendix D. The main idea is to use the Jacobian [5]. In fact, it solves a more general model than $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$.

Transcendence basis. Polynomials $T_1, \ldots, T_m$ are called *algebraically dependent* if there exists a nonzero *annihilator* A s.t. $A(T_1, \ldots, T_m) = 0$. *Transcendence degree* is the size of the largest subset $S \subseteq \{T_1, \ldots, T_m\}$ that is algebraically independent. Then S is called a *transcendence basis.*

▶ **Problem 1.** *Let $\{T_i \mid i \in [m]\}$ be $\Pi\Sigma\Pi^{[\delta]}$ circuits of (syntactic) degree at most d and size s. Let the transcendence degree of T_i's, $\mathrm{trdeg}_{\mathbb{F}}(T_1, \ldots, T_m) = k \ll s$. Further, $C(x_1, \ldots, x_m)$ be a circuit of $(\mathrm{size} + \deg) < s'$. Design a blackbox-PIT algorithm for $C(T_1, \ldots, T_m)$.*

Trivially, $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ is a very special case of the above setting. Let $\boldsymbol{T} := \{T_1, \ldots, T_m\}$. Let $\boldsymbol{T}_k := \{T_1, \ldots, T_k\}$ be a transcendence basis. For $T_i = \prod_j g_{ij}$, we denote the set $L(T_i) := \{g_{ij} \mid j\}$.

We want to find an explicit homomorphism $\Psi : \mathbb{F}[\boldsymbol{x}] \to \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ s.t. $\Psi(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$ is of a 'nice' form. In the image we fix $\boldsymbol{x}$ suitably, to get a composed map $\Psi' : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_1, z_2]$ s.t. $\mathrm{rk}_{\mathbb{F}(\boldsymbol{x})}\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{z})}\Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$. Then, we can extend this map to $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, \boldsymbol{y}, t]$ s.t. $x_i \mapsto (\sum_{j=1}^{k} y_j t^{ij}) + \Psi'(x_i)$, which is *faithful* [5, Lemma 2.7]; see Lemma 21. We show that the map Φ can be efficiently constructed using a scaling and shifting map (Ψ) which is eventually fixed by the hitting set (H' defining Ψ') of a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit. Overall, $\Phi(f)$ is a $k+3$-variate polynomial for which a trivial hitting set exists.

Wlog, $\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T})$ is full rank with respect to the variable set $\boldsymbol{x}_k = (x_1, \ldots, x_k)$. Thus, by assumption, $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \neq 0$ (for notation, see Section 2). We want to construct a Ψ s.t. $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))$ has an 'easier' PIT. We have the following identity [5, Eqn. 3.1], from the linearity of the determinant, and the simple observation that $\partial_x(T_i) \;=\; T_i \cdot \left(\sum_j \partial_x(g_{ij})/g_{ij}\right)$, where $T_i \;=\; \prod_j g_{ij}$:

$$J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \;=\; \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \left(\frac{T_1\ldots T_k}{g_1\ldots g_k}\right)\cdot J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)\;. \tag{5}$$

The homomorphism Ψ. Define $\Psi : \mathbb{F}[\boldsymbol{x}] \to \mathbb{F}[\boldsymbol{x}, z_1, z_2]$ as $x_i \mapsto z_1 \cdot x_i + \Psi_1(x_i)$, where $\Psi_1 : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$, is a *sparse-PIT* map. The importance of Ψ_1 is to ensure that $\Psi_1(g) \neq 0$, $\forall g \in \bigcup_i L(T_i)$. As $\deg(g) \le \delta$, $\mathrm{sp}(g) \le \binom{n+\delta}{\delta}$, . Thus, [49] (Theorem 10) gives the upper bound:

$$\deg_{z_2}(\Psi(g)) \le \delta \cdot \left(\binom{n+\delta}{\delta}\cdot n\cdot \log\delta\right)^2 =: D_1.$$

Denote the ring $\mathsf{R}[\boldsymbol{x}]$ where $\mathsf{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^D\rangle$, and $D := k\cdot(d-1)+1$. Being 1-1, Ψ is clearly a non-zero preserving map. Moreover,

▷ Claim 5. $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \;=\; 0 \iff \Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) \;=\; 0$, over $\mathsf{R}[\boldsymbol{x}]$.

Proof. As $\deg(T_i) \le d$, each entry of the matrix can be of degree at most $d-1$; therefore $\deg(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) \le k(d-1) = D-1$. Thus, $\deg_{z_1}(\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))) < D$. Hence, the conclusion. ◀

Eqn. (5) implies that

$$\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) \;=\; \Psi(T_1\cdots T_k)\cdot \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))}{\Psi(g_1\ldots g_k)}\;. \tag{6}$$

As T_i has product fanin s, the top-fanin in the sum in Eqn. (6) can be at most s^k. Then define,

$$\widetilde{F} \;:=\; \sum_{g_1\in L(T_1),\ldots,g_k\in L(T_k)} \frac{\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))}{\Psi(g_1\ldots g_k)}\;,\quad \text{over } \mathsf{R}[\boldsymbol{x}]. \tag{7}$$

Well-definability of $\widetilde{F}$. Note that,

$$\Psi(g_i) \equiv \Psi_1(g_i) \bmod z_1 \neq 0 \implies 1/\Psi(g_1\cdots g_k) \in \mathbb{F}(z_2)[[\boldsymbol{x}, z_1]].$$

Thus, RHS is an element in $\mathbb{F}(z_2)[[\boldsymbol{x}, z_1]]$ and taking mod z_1^D it is in $\mathsf{R}[\boldsymbol{x}]$. We remark that instead of minimally reducing mod z_1^D, we will work with an $F \in \mathbb{F}(z_2)[z_1, \boldsymbol{x}]$ such that $F = \tilde{F}$ over $\mathsf{R}[\boldsymbol{x}]$. Further, we ensure that the degree of $\boldsymbol{z}$ is polynomially bounded.

▷ Claim 6. Over $\mathsf{R}[\boldsymbol{x}]$, $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k)) = 0 \iff F = 0$.

Proof sketch. This follows from the invertibility of $\Psi(T_1\cdots T_k)$ in $R[\boldsymbol{x}]$. ◀

The hitting set H'. By $J_{\boldsymbol{x}_k}(\boldsymbol{T}_k) \neq 0$, and Claims 5-6, we have $F \neq 0$ over $\mathsf{R}[\boldsymbol{x}]$. We want to find $H' \subseteq \mathbb{F}^n$, s.t. $\Psi(J_{\boldsymbol{x}_k}(\boldsymbol{T}_k))|_{\boldsymbol{x}=\boldsymbol{\alpha}} \neq 0$, for some $\boldsymbol{\alpha} \in H'$ (which will ensure the rank-preservation). Towards this, we will show (below) that F has $s^{O(\delta k)}$-size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit over $\mathsf{R}[\boldsymbol{x}]$. Next, Theorem 24 provides the hitting set H' in time $s^{O(\delta^2 k\log s)}$.

▷ Claim 7 (Main size bound). $F \in \mathsf{R}[\boldsymbol{x}]$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size $(s3^\delta)^{O(k)}$.

The proof studies the two parts of Eqn. (7)—

1. The numerator $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))$ has $O(3^\delta 2^k k!ks)$-size $\Sigma\wedge\Sigma\Pi^{[\delta-1]}$-circuit (see Lemma 8), and
2. $1/\Psi(g_1\cdots g_k)$, for $g_i \in L(T_i)$ has $(s3^\delta)^{O(k)}$-size $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit; both over $\mathsf{R}[\boldsymbol{x}]$ (see Lemma 9).

▶ **Lemma 8** (Numerator size). $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma\wedge\Sigma\Pi^{[\delta-1]}$ *of size* $O(3^\delta\, 2^k k\, k!s) =: s_2$.

Proof sketch. One can show that $J_{\boldsymbol{x}_k}(g_1,\ldots,g_k) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$ of size $O(k!ks)$, where $g_i \in L(T_i)$ (Claim 22): this basically follows from the determinant expansion which has fanin $k!$ and the degree at the bottom is $\leq \delta-1$ because of the derivative. Moreover, for a $g \in \Sigma\Pi^{[\delta-1]}$, we have $\Psi(g) \in \Sigma\Pi^{[\delta-1]}$ of size at most $3^\delta\cdot\mathrm{size}(g)$, over $\mathsf{R}[\boldsymbol{x}]$ (Claim 23): this follows from the fact that $\boldsymbol{x}^{\boldsymbol{e}}$ (where $|\boldsymbol{e}|_0 \leq \delta$), after shift, can produce at most $\prod(e_i+1) \leq e^\delta$ many monomials (for large n). Combining these, one concludes $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$, of size $O(3^\delta\, k!ks)$. We *convert* the Π-gate to $\wedge$ gate using waring identity (Lemma 14) which blowsup the size by a multiple of 2^{k-1}. Thus, $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma\wedge\Sigma\Pi^{[\delta-1]}$ of size $O(3^\delta\, 2^k k\, k!s)$. ◀

By power series expansion of expressions like $1/(1-a\cdot z_1)$, one can conclude that $1/\Psi(g)$ has a small $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit, which would further imply the same for $1/\Psi(g_1\cdots g_k)$ (see below).

▶ **Lemma 9** (Denominator size). *Let* $g_i \in L(T_i)$*. Then,* $1/\Psi(g_1\cdots g_k)$ *can be computed by a* $\Sigma\wedge\Sigma\Pi^{[\delta]}$*-circuit of size* $s_1 := (s3^\delta)^{O(k)}$*, over* $\mathsf{R}[\boldsymbol{x}]$.

Proof. Let $g \in L(T_i)$ for some i. Assume, $\Psi(g) = A - z_1\cdot B$, for some $A \in \mathbb{F}[z_2]$ and $B \in \mathsf{R}[\boldsymbol{x}]$ of degree δ, with $\mathrm{size}(B) \leq 3^\delta\cdot s$, from Claim 23. Note that, over $\mathsf{R}[\boldsymbol{x}]$,

$$\frac{1}{\Psi(g)} \;=\; \frac{1}{A(1-\frac{B}{A}\cdot z_1)} \;=\; \frac{1}{A}\cdot\sum_{i=0}^{D-1}\left(\frac{B}{A}\right)^i\cdot z_1^i\,. \tag{8}$$

As, $\mathrm{size}(B^i)$ has a trivial $\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathsf{R}[\boldsymbol{x}]$) of size $\leq 3^\delta\cdot s + i$; summing over $i \in [D-1]$, the overall size is at most $D\cdot 3^\delta\cdot s + O(D^2)$. As $D < k\cdot d$, we conclude that $1/\Psi(g)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$ of size $\mathsf{poly}(s\cdot k\cdot d3^\delta)$, over $\mathsf{R}[\boldsymbol{x}]$. Multiplying k-many such products directly gives an upper bound of $(s\cdot 3^\delta)^{O(k)}$, using Lemma 15 (basically, waring identity). ◀

Proof of Claim 7. Combining Lemmas 8-9, observe that $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k))/\Psi(g_1\cdots g_k)$ has $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit of size at most $(s_1\cdot s_2)^2 = (s\cdot 3^\delta)^{O(k)}$, over $\mathsf{R}[\boldsymbol{x}]$, using Lemma 15. Summing up at most s^k many terms (by defn. of F), the size still remains $(s\cdot 3^\delta)^{O(k)}$. ◀

Degree bound. As, syntactic degree of T_i are bounded by d, and Ψ maintain $\deg_{\boldsymbol{x}} = \deg_{z_1}$, we must have $\deg_{z_1}(\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) = \deg_{\boldsymbol{x}}(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \leq D-1$. Similarly, by assumption $\deg_{z_2}(\Psi(g)) \leq D_1 := \mathsf{poly}(n^\delta)$, and thus $\deg_{z_2}(\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \leq D_1\cdot k$. Note that, Lemma 8 actually works over $\mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$ and thus there is no additional degree-blow up (in $\boldsymbol{z}$). However, there is some degree blowup in Lemma 9, due to Eqn. (8).

Note that Eqn. (8) shows that over $\mathsf{R}[\boldsymbol{x}]$,

$$\frac{1}{\Psi(g)} = \left(\frac{1}{A^D}\right)\cdot\left(\sum_{i=0}^{D-1}A^{D-1-i}z_1^i\cdot B^i\right) =: \frac{p(\boldsymbol{x},\boldsymbol{z})}{q(z_2)},$$

where $q(z_2) = A^D$. We think of $p \in \mathbb{F}[\boldsymbol{x}, \boldsymbol{z}]$ and $q \in \mathbb{F}[z_2]$. It follows that $\deg_{z_2}(q) \le D_1 \cdot D$. Also, $\deg_{z_1}(\Psi(g)) \le \delta$ implies $\deg_{z_1}(p) \le \deg_{z_1}((B\,z_1)^{D-1}) \le \delta \cdot (D-1)$. Since, $\deg_{z_2}(\Psi(g)) \le D_1$, by assumption, $\deg_{z_2}(p) \le \max_i \deg_{z_2}(A^{D-1-i} \cdot B^i) \le D_1 \cdot (D-1)$.

Finally, denote $1/\Psi(g_1 \cdots g_k) =: P_{g_1,\ldots,g_k}/Q_{g_1,\ldots,g_k}$, over $\mathsf{R}[\boldsymbol{x}]$. This is just multiplying k-many (p/q)'s; implying a degree blowup by a multiple of k. In particular,

- $\deg_{z_1}(P_{(\cdot)}) \le \delta \cdot k \cdot (D-1)$,
- $\deg_{z_2}(P_{(\cdot)}) \le D_1 \cdot (D-1) \cdot k$, and
- $\deg_{z_2}(Q_{(\cdot)}) \le D_1 \cdot D \cdot k$.

Thus, in Eqn. (7), summing up s^k-many terms gives an expression (over $\mathsf{R}[\boldsymbol{x}]$):

$$F \;=\; \sum_{g_1 \in L(T_1),\ldots,g_k \in L(T_k)} \Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \cdot \left(\frac{P_{g_1,\ldots,g_k}}{Q_{g_1,\ldots,g_k}}\right) \;=:\; \frac{P(\boldsymbol{x},\boldsymbol{z})}{Q(z_2)}\;.$$

Verify that $Q \in \mathbb{F}[z_2]$ is of degree at most $s^k \cdot D_1 \cdot D \cdot k = s^{O(k)} \cdot \mathsf{poly}(n^\delta)$ (since $k, d < s$). A similar bound also holds for $\deg_{z_2}(P)$. The degree of z_1 also remains bounded by

$$\max_{g_i \in L(T_i), i \in [k]} \deg_{z_1}(P_{g_1,\ldots,g_k}) + \delta k \le \mathsf{poly}(s).$$

Using the degree bounds, we finally have $P \in \mathbb{F}[\boldsymbol{x}, \boldsymbol{z}]$ as a $\Sigma\wedge\Sigma\Pi^{[\delta]}$-circuit (over $\mathbb{F}(\boldsymbol{z})$) of size $n^{O(\delta)}\,(s3^\delta)^{O(k)} = 3^{O(\delta k)} s^{O(k+\delta)} =: s_3$.

We want to *construct* a set $H' \subseteq \mathbb{F}^n$ such that the action $P(H', \boldsymbol{z}) \neq 0$. Using [25] (Theorem 24), we conclude that it has $s^{O(\delta \log s_3)} = s^{O(\delta^2 k \log s)}$ size hitting set which is constructible in a similar time. Hence, the construction of Φ follows, making $\Phi(f)$ a $k+3$ variate polynomial. Finally, by the obvious degree bounds of $\boldsymbol{y}, \boldsymbol{z}, t$ from the definition of Φ, we get the blackbox PIT algorithm with time-complexity $s^{O(\delta^2 k \log s)}$; finishing Theorem 2b.

We could also give the final hitting set for the general problem.

Solution to Problem 1. We know that $C(T_1,\ldots,T_m) = 0 \iff E := \Phi(C(T_1,\ldots,T_m)) = 0$. Since, H' can be constructed in $s^{O(\delta^2\,k\,\log s)}$-time, it is trivial to find hitting set for $E|_{H'}$ (which is just a $k+3$-variate polynomial with the aforementioned degree bounds). The final hitting set for E can be constructed in $s'^{O(k)} \cdot s^{O(\delta^2\,k\,\log s)}$-time. ◀

▶ Remark. **1.** As Jacobian Criterion (Fact 2) holds when the characteristic is $> d^{\mathrm{trdeg}}$, it is easy to conclude that our theorem holds for all fields of char $> d^k$.

2. The above proof gives an efficient reduction from blackbox PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits to $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits. In particular, a poly-time hitting set for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits would put PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ in P.

3. Also, DiDI-technique (of Theorem 1) directly gives a blackbox algorithm, but the complexity is *exponentially* worse (in terms of k in the exponent) for its recursive blowups.

4 Conclusion

This work introduces the powerful DiDI-technique and solves three open problems in PIT for depth-4 circuits, namely $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ (blackbox) and $\Sigma^{[k]}\Pi\Sigma\wedge$ (both whitebox and blackbox). Here are some immediate questions of interest which require rigorous investigation.

1. Can the exponent in Theorem 1 be improved to $O(k)$? Currently, it is exponential in k.
2. Can we improve Theorem 2b to $s^{O(\log \log s)}$ (like in Theorem 2a)?
3. Can we design a polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$?
4. Design a poly-time PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits (i.e. unbounded top-fanin)?
5. Can we solve PIT for $\Sigma^{[k]}\Pi\Sigma\wedge^{[2]}$ in *sub*exponential-time?
6. Can we design a subexponential-time PIT for rational functions of the form $\Sigma\,(1/\Sigma\wedge\Sigma)$ or $\Sigma\,(1/\Sigma\Pi)$ (for *un*bounded top-fanin)?

References

1 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 92–105. Springer, 2005.

2 Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. Preliminary version in Symposium on Theory of Computing, 2018 (STOC'18). doi:10.1073/pnas.1901272116.

3 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015.

4 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

5 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-D occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM Journal on Computing*, 45(4):1533–1562, 2016. Preliminary version in 44^{th} Symposium on Theory of Computing, 2018 (STOC'12).

6 Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proceedings of the 45^{th} Annual ACM symposium on Theory of computing (STOC'13)*, pages 321–330, 2013.

7 Manindra Agrawal and V Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 67–75. IEEE, 2008.

8 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 Transactions on Computation Theory (TOCT)*, 10(1):1–30, 2018. Preliminary version in the IEEE 31^{st} Computational Complexity Conference (CCC'16).

9 Robert Andrews. Algebraic Hardness Versus Randomness in Low Characteristic. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *LIPIcs*, pages 37:1–37:32. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

10 Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM (JACM)*, 45(3):501–555, 1998.

11 Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998. Preliminary version in 33^{rd} Annual Symposium on Foundations of Computer Science (FOCS'92).

12 Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Information and Computation*, 222:2–19, 2013. Preliminary version in 38^{th} International Colloquium on Automata, Languages and Programming (ICALP'11).

13 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the 20^{th} Annual ACM symposium on Theory of computing (STOC'88)*, pages 301–309, 1988.

14 Pranav Bisht and Nitin Saxena. Poly-time blackbox identity testing for sum of log-variate constant-width ROABPs. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 27, page 42, 2020.

15 Enrico Carlini, Maria Virginia Catalisano, and Anthony V. Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of Algebra*, 370:5 – 14, 2012. URL: `http://www.sciencedirect.com/science/article/pii/S0021869312003730`, `doi:https://doi.org/10.1016/j.jalgebra.2012.07.028`.

16 Prerona Chatterjee, Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the Existence of Algebraically Natural Proofs. In *IEEE* 61^{st} *Annual Symposium on Foundations of Computer Science (FOCS'20)*, 2020.

17 Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs randomness for bounded depth arithmetic circuits. In 33^{rd} *Computational Complexity Conference (CCC'18)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

18 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978.

19 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proceedings of the* 50^{th} *Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, pages 1152–1165, 2018.

20 Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 23:1–23:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.

21 Zeev Dvir, Rafael Mendes De Oliveira, and Amir Shpilka. Testing equivalence of polynomials under shifts. In *International Colloquium on Automata, Languages, and Programming*, pages 417–428. Springer, 2014.

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

23 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM Journal on Computing*, 39(4):1279–1293, 2010. Preliminary version in Proceedings of the 40^{th} Annual ACM symposium on Theory of computing (STOC'08).

24 Stephen Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. *SIAM Journal on Computing*, 62(3):109–115, 2019. Preliminary version in Proceedings of the 48^{th} Annual ACM symposium on Theory of Computing (STOC'16).

25 Michael A Forbes. Deterministic divisibility testing via shifted partial derivatives. In *Proceedings of the* 56^{th} *Annual Symposium on Foundations of Computer Science (FOCS'15)*, pages 451–465. IEEE, 2015.

26 Michael A Forbes, Sumanta Ghosh, and Nitin Saxena. Towards blackbox identity testing of log-variate circuits. In 45^{th} *International Colloquium on Automata, Languages, and Programming (ICALP'18)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

27 Michael A Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the* 46^{th} *Annual ACM symposium on Theory of computing (STOC'14)*, pages 867–875, 2014.

28 Michael A Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In 54^{th} *Annual Symposium on Foundations of Computer Science (FOCS'13)*, pages 243–252, 2013.

29 Michael A Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14:1–45, 2018. Preliminary version in Proceedings of the 49^{th} Annual ACM SIGACT Symposium on Theory of Computing (STOC'19).

30 Abhibhav Garg and Nitin Saxena. Special-case algorithms for blackbox radical membership, Nullstellensatz and transcendence degree. In *Proceedings of the* 45^{th} *International Symposium on Symbolic and Algebraic Computation*, pages 186–193, 2020.

31 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In 57^{th} *Annual Symposium on Foundations of Computer Science (FOCS'16)*, pages 109–117. IEEE, 2016.

32 Joshua A Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. Preliminary version in the IEEE $29th$ Computational Complexity Conference (CCC'14).

33 Zeyu Guo. Variety Evasive Subspace Families. In *Manuscript*, 2021.

34 Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness: Treading the Borders. In 60^{th} *IEEE Annual Symposium on Foundations of Computer Science (FOCS'19)*, pages 147–157. IEEE Computer Society, 2019.

35 Ankit Gupta. Algebraic Geometric Techniques for Depth-4 PIT & Sylvester-Gallai Conjectures for Varieties. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 21, page 130, 2014.

36 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. *SIAM Journal on Computing*, 45(3):1064–1079, 2016. 54^{th} Annual Symposium on Foundations of Computer Science (FOCS'13).

37 Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs. *Theory of Computing*, 13(2):1–21, 2017. Preliminary version in the 31^{st} Computational Complexity Conference (CCC'16). `doi:10.4086/toc.2017.v013a002`.

38 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Computational Complexity*, 26(4):835–880, 2017. Preliminary version in the IEEE 30^{th} Computational Complexity Conference (CCC'15).

39 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute. In *Proceedings of the* 12^{th} *annual ACM symposium on Theory of computing (STOC'80)*, pages 262–272, 1980.

40 Maurice Jansen, Youming Qiao, and Jayalal Sarma. Deterministic Black-Box Identity Testing π-Ordered Algebraic Branching Programs. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2010*, volume 8 of *LIPIcs*, pages 296–307. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.

41 A Grochow Joshua, D Mulmuley Ketan, and Qiao Youming. Boundaries of VP and VNP. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 34:1–34:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

42 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the Proceedings of the 35^{th} Annual ACM symposium on Theory of computing (STOC'03).

43 Zohar S Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM Journal on Computing*, 42(6):2114–2131, 2013. Preliminary version in the Proceedings of the 42^{nd} ACM symposium on Theory of computing (STOC'10).

44 Zohar S Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In 24^{th} *Annual IEEE Conference on Computational Complexity (CCC'09)*, pages 274–285. IEEE, 2009.

45 Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333, 2011. Preliminary version in the 23^{rd} Annual IEEE Conference on Computational Complexity (CCC'08).

46 Neeraj Kayal, Pascal Koiran, Timothée Pecatte, and Chandan Saha. Lower bounds for sums of powers of low degree univariates. In *International Colloquium on Automata, Languages, and Programming (ICALP'15)*, pages 810–821. Springer, 2015.

47 Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007. Preliminary version in the 21^{st} Computational Complexity Conference (CCC'06).

48 Adam Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(1):185–206, 2006. Preliminary version in the 16^{th} Annual Conference on Learning Theory (COLT'03).

49 Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the* 33^{rd} *Annual ACM symposium on Theory of computing (STOC'01)*, pages 216–223, 2001.

50 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012.

51 Pascal Koiran, Natacha Portier, and Sébastien Tavenas. A Wronskian approach to the real τ-conjecture. *Journal of Symbolic Computation*, 68:195–214, 2015.

52 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *IEEE* 29^{th} *Conference on Computational Complexity (CCC'14)*, pages 169–180. IEEE, 2014.

53 Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. If VNP is hard, then so are equations for it. *Preprint avilable at arXiv:2012.07056*, 2020.

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

55 Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the* 30^{th} *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646, 2019.

56 Mrinal Kumar and Shubhangi Saraf. Sums of Products of Polynomials in Few Variables: Lower Bounds and Polynomial Identity Testing. In 31^{st} *Conference on Computational Complexity, CCC 2016*, volume 50 of *LIPIcs*, pages 35:1–35:29. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

57 Mrinal Kumar and Shubhangi Saraf. Arithmetic Circuits with Locally Low Algebraic Rank. *Theory Comput.*, 13(1):1–33, 2017. Preliminary version in the 31^{st} Conference on Computational Complexity (CCC'16).

58 Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Chic. J. Theor. Comput. Sci.*, 2:1–19, 2019.

59 László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory (FCT'79)*, volume 79, pages 565–574, 1979.

60 Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM (JACM)*, 39(4):859–868, 1992.

61 Partha Mukhopadhyay. Depth-4 identity testing and Noether's normalization lemma. In *International Computer Science Symposium in Russia (CSR'16)*, pages 309–323. Springer, 2016.

62 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. Preliminary version in the Proceedings of the 19^{th} Annual ACM symposium on Theory of Computing (STOC'87).

63 Ketan D Mulmuley. Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma. In *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS'12)*, pages 629–638. IEEE, 2012.

64 Ketan D Mulmuley. The GCT program toward the P vs. NP problem. *Communications of the ACM*, 55(6):98–107, 2012.

65 Ivan Niven. Formal power series. *The American Mathematical Monthly*, 76(8):871–889, 1969. URL: `http://www.jstor.org/stable/2317940`.

66 Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

67 Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low-algebraic-rank circuits. *Computational Complexity*, 27(4):617–670, 2018. Preliminary version in the 41^{st} International Symposium on Mathematical Foundations of Computer Science (MFCS'16).

68 Shir Peleg and Amir Shpilka. A generalized Sylvester-Gallai type theorem for quadratic polynomials. In 35^{th} *Computational Complexity Conference (CCC'20)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2020.

69 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\sum^{[3]}\prod\sum\prod^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials. In 53^{rd} *Annual ACM symposium on Theory of computing (STOC'21)*, 2021.

70 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. Preliminary version in the 19^{th} IEEE Annual Conference on Computational Complexity (CCC'04).

71 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. A case of depth-3 identity testing, sparse factorization and duality. *Computational Complexity*, 22(1):39–69, 2013.

72 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2019.

73 Ramprasad Saptharishi. Private communication, 2019.

74 Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. *Combinatorica*, 38(5):1205–1238, 2018. Preliminary version in the Proceedings of the 43^{rd} Annual ACM symposium on Theory of computing (STOC'11).

75 Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *International Colloquium on Automata, Languages, and Programming (ICALP'08)*, pages 60–71. Springer, 2008.

76 Nitin Saxena. Progress on Polynomial Identity Testing. *Bulletin of the EATCS*, 99:49–79, 2009.

77 Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in Computational Complexity*, pages 131–146. Springer, 2014.

78 Nitin Saxena and Comandur Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM journal on computing*, 40(1):200–224, 2011. Preliminary version in the 24^{th} IEEE Conference on Computational Complexity (CCC'09).

79 Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM Journal on Computing*, 41(5):1285–1298, 2012. Preliminary version in the 43^{rd} Annual ACM symposium on Theory of computing (STOC'11).

80 Nitin Saxena and Comandur Seshadhri. From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):1–33, 2013. Preliminary version in the 51^{st} Annual IEEE Symposium on Foundations of Computer Science (FOCS'10).

81 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

82 Adi Shamir. IP= PSPACE. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.

83 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM Journal on Computing*, 38(6):2130–2161, 2009. Preliminary version in the Proceedings of the 39^{th} Annual ACM symposium on Theory of Computing (STOC 2007).

84 Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. In *Proceedings of the* 51^{st} *Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*, pages 1203–1214, 2019.

85 Amir Shpilka and Amir Yehudayoff. *Arithmetic circuits: A survey of recent results and open questions*. Now Publishers Inc, 2010.

86 Amit Kumar Sinhababu. *Power series in complexity: Algebraic Dependence, Factor Conjecture and Hitting Set for Closure of VP*. PhD thesis, PhD thesis, Indian Institute of Technology Kanpur, 2019.

87 Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the* 11^{th} *Annual ACM symposium on Theory of computing (STOC'79)*, pages 249–261, 1979.

88 Wolmer Vasconcelos. *Computational methods in commutative algebra and algebraic geometry*, volume 2. Springer Science & Business Media, 2004.

89 Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979.

A Basic tools from algebraic complexity

There have been a lot of work on sparse-PIT, for details see [13, 49] and references therein. Eventually, there is a poly-time hitting set, for a proof see [76, Thm. 2.1]

▶ **Theorem 10** ([49]). *Let* $p(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ *with individual degree at most* d *and sparsity at most* m. *Then, there exists* $1 \leq r \leq (mn \log d)^2$, *such that* $p(y, y^d, \ldots, y^{d^{n-1}}) \neq 0, \bmod y^r - 1$.

An **ABP** is a layered directed acyclic graph with $q+1$ many layers of vertices $\{V_0, \ldots, V_q\}$ and a source a and a sink b such that all the edges in the graph only go from a to V_0, V_{i-1} to V_i for any $i \in [q]$, and V_q to b. The edges have *uni*variate polynomials as their weights. The ABP is said to compute the polynomial

$$f(\boldsymbol{x}) \;=\; \sum_{p \in \mathrm{paths}(a,b)} \prod_{e \in p} W(e) \;,$$

where $W(e)$ is the weight of the edge e. The ABP has width-w if $|V_i| \leq w$, $\forall i \in \{0, \ldots, q\}$. Formally, it computes polynomials of the form $A^T(\prod_{i \in [q]} D_i)B$, where $A, B \in \mathbb{F}^{w \times 1}[\boldsymbol{x}]$, and $D_i \in \mathbb{F}^{w \times w}[\boldsymbol{x}]$, where entries are univariate polynomials.

▶ **Definition 11** (Read-once oblivious ABP (ROABP)). *An ABP is called a* read-once oblivious ABP (ROABP) *if the edge weights are univariate polynomials in* distinct *variables across layers. Formally, there is a permutation* π *on the set* $[q]$ *such that the entries in the* i*-th matrix* D_i *are univariate polynomials over the variable* $x_{\pi(i)}$, *i.e., they come from the polynomial ring* $\mathbb{F}[x_{\pi(i)}]$.

A polynomial $f(x)$ is said to be computed by width-w ROABPs in *any order*, if for every permutation σ of the variables, there exists a width-w ROABP in the variable order s that computes the polynomial $f(\boldsymbol{x})$. There have been quite a few results on blackbox PIT for ROABPs [28, 27, 37] and the current best known algorithm works in quasipolynomial time.

▶ **Theorem 12** ([37]). *For* n*-variate, individual-degree-*d *polynomials computed by width-*w *ROABPs in any order, a hitting set of size* $(ndw)^{O(\log \log w)}$ *can be constructed.*

B Details for Section 3.1

Here is an important lemma which shows that coefficient of y^e of a polynomial $f(\boldsymbol{x}, y) \in \mathbb{F}[\boldsymbol{x}, y]$, computed by a $\Sigma\wedge\Sigma\wedge$ circuit, can be computed by a small $\Sigma\wedge\Sigma\wedge$ circuit.

▶ **Lemma 13** (Coefficient extraction). *Let* $f(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$ *be computed by a* $\Sigma\wedge\Sigma\wedge$ *circuit of size* s *and degree* d. *Then,* $\mathrm{coef}_{y^e}(f) \in \mathbb{F}[\boldsymbol{x}]$ *can be computed by a small* $\Sigma\wedge\Sigma\wedge$ *circuit of size* $O(sd)$, *over* $\mathbb{F}[\boldsymbol{x}]$.

Proof sketch. Let, $f = \sum_i \alpha_i \cdot g_i^{e_i}$. Of course, $e_i \leq s$ and $\deg_y(f) \leq d$. Thus, write $f = \sum_{i=0}^{d} f_i \cdot y^i$, where $f_i \in \mathbb{F}[\boldsymbol{x}]$. We can interpolate on $d+1$-many distinct points $y \in \mathbb{F}$ and conclude that f_i has a $\Sigma\wedge\Sigma\wedge$ circuit of size at most $O(sd)$. ◀

The next identity gives us a way to write a product of a few powers as a sum of powers, using simple interpolation. For a more algebraic proof, see [15, Proposition 4.3].

▶ **Lemma 14** (Waring Identity for a monomial). *Let $M = x_1^{b_1} \cdots x_k^{b_k}$, where $1 \le b_1 \le \ldots \le b_k$, and roots of unity $\mathcal{Z}(i) := \{z \in \mathbb{C} : z^{b_i+1} = 1\}$. Then,*

$$M = \sum_{\varepsilon(i) \in \mathcal{Z}(i): i=2,\cdots,k} \gamma_{\varepsilon(2),\ldots,\varepsilon(k)} \cdot (x_1 + \varepsilon(2)x_2 + \ldots + \varepsilon(k)x_k)^d \ ,$$

where $d := \deg(M) = b_1 + \ldots + b_k$, and $\gamma_{\varepsilon(2),\ldots,\varepsilon(k)}$ are scalars ($\mathrm{rk}(M) := \prod_{i=2}^k (b_i + 1)$ many).

Remark. We actually need not work with $\mathbb{F} = \mathbb{C}$. We can go to a small extension (at most d^k), for a monomial of degree d, to make sure that $\varepsilon(i)$ exists.

The next lemma shows that $\Sigma\wedge\Sigma\wedge$ is *closed* under multiplication.

▶ **Lemma 15.** *Let $f_i(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$, of syntactic degree $\le d_i$, be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s_i, for $i \in [k]$ (wrt $\boldsymbol{x}$). Then, $f_1 \cdots f_k$ has $\Sigma\wedge\Sigma\wedge$ circuit of size $O((d_2+1)\cdots(d_k+1) \cdot s_1 \cdots s_k)$.*

Proof. Let $f_i = \sum_j f_{ij}^{e_{ij}}$; by assumption $e_{ij} \le d_i$ (by assumption). Using Lemma 14, $f_{1j_1}^{e_{1j_1}} \cdots f_{kj_k}^{e_{kj_k}}$ has size at most $(d_2+1)\cdots(d_k+1) \cdot \left(\sum_{i\in[k]} \mathrm{size}(f_{ij_i})\right)$, for indices $j_1, \ldots, j_k$. Summing up for all $s_1 \cdots s_k$ many products (atmost) gives the upper bound. ◀

The next lemma shows that $\Sigma\wedge\Sigma\wedge$ is *closed* under differentiation.

▶ **Lemma 16** (Differentiation). *Let $f(\boldsymbol{x}, y) \in \mathbb{F}[y][\boldsymbol{x}]$ be computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s and degree d. Then, $\partial_y(f)$ can be computed by a small $\Sigma\wedge\Sigma\wedge$ circuit of size $O(sd^2)$, over $\mathbb{F}[y][\boldsymbol{x}]$.*

Proof sketch. Lemma 13 shows that each f_e has $O(sd)$ size circuit where $f = \sum_e f_e\, y^e$. Doing this for each $e \in [0, d]$ gives a blowup of $O(sd^2)$. ◀

The next lemma shows that non-negative valuation corresponds to a power-series.

▶ **Lemma 17** (Valuation). *Consider a polynomial $f \in \mathbb{F}(\boldsymbol{x}, y)$ such that $\mathrm{val}_y(f) \ge 0$. Then, $f \in \mathbb{F}(\boldsymbol{x})[[y]] \bigcap \mathbb{F}(\boldsymbol{x}, y)$.*

Proof sketch. Let $f = g/h$, where $g, h \in \mathbb{F}[\boldsymbol{x}, y]$. Now, $\mathrm{val}_y(f) \ge 0$, implies $\mathrm{val}_y(g) \ge \mathrm{val}_y(h)$. Let $\mathrm{val}_y(g) = d_1$ and $\mathrm{val}_y(h) = d_2$, where $d_1 \ge d_2 \ge 0$. Write $g = y^{d_1} \cdot \tilde{g}$ and $h = y^{d_2} \cdot \tilde{h}$. Write, $\tilde{h} = h_0 + h_1\, y + h_2\, y^2 + \ldots + h_d\, y^d$, for some d. Note that $h_0 \ne 0$. Thus,

$$\begin{aligned} f &= y^{d_1-d_2} \cdot \tilde{g}/(h_0 + h_1\, y + \ldots + h_d\, y^d) \\ &= y^{d_1-d_2} \cdot (\tilde{g}/h_0) \cdot (1 + (h_1/h_0)\, y + \ldots + (h_d/h_0)\, y^d)^{-1} \ \in \mathbb{F}(\boldsymbol{x})[[y]] \,. \end{aligned}$$

The last conclusion follows by the inverse identity in the power-series ring. ◀

Using duality trick [75] and PIT results from [70, 37], one can design efficient PIT algorithm for $\Sigma\wedge\Sigma\wedge$ circuits:

▶ **Lemma 18** (PIT for $\Sigma\wedge\Sigma\wedge$-circuits). *Let $P \in \Sigma\wedge\Sigma\wedge$ of size s. Then, there exists a* poly(s) *(respec. $s^{O(\log\log s)}$) time whitebox (respec. blackbox) PIT for the same.*

Proof sketch. We show that any $g(\boldsymbol{x})^e = (g_1(x_1) + \ldots + g_n(x_n))^e$, where $\deg(g_i) \le s$ can be written as $\sum_j h_{j1}(x_1) \cdots h_{jn}(x_n)$, for some $h_{j\ell} \in \mathbb{F}[x_\ell]$ of degree at most es. Define, $G := (y + g_1) \cdots (y + g_n) - y^n$. In its e-th power, notice that the leading-coefficient is $\mathrm{coef}_{y^{e(n-1)}}(G^e) = g^e$. So, interpolate on $e(n-1)+1$ many points ($y = \beta_i \in \mathbb{F}$) to get

$$\mathrm{coef}_{y^{e(n-1)}}(G^e) \;=\; \sum_{i=1}^{e(n-1)+1} \alpha_i \, G^e(\beta_i) \; .$$

Now, expand $G^e(\beta_i) = ((\beta_i + g_1) \cdots (\beta_i + g_n) - \beta_i^n)^e$, by binomial expansion (without expanding the inner n-fold product). The top-fanin can be atmost $s \cdot (e+1) \cdot (e(n-1)+1) = O(se^2 n)$. The individual degrees of the intermediate univariates can be at most es. Thus, it can be computed by an ROABP (of *any order*) of size at most $O(s^2 e^3 n)$.

Now, if $f = \sum_{j \in [s]} f_j^{e_j}$ is computed by a $\Sigma\wedge\Sigma\wedge$ circuit of size s, then clearly, f can also be computed by an ROABP (of any order) of size at most $O(s^6)$. So, the whitebox PIT follows from [70], while the blackbox PIT follows from Theorem 12. ◀

For the time-complexity bound, we need optimization of the following function:

▶ **Lemma 19.** *Let $k \in \mathbb{N}$, and $h(x) := x(k-x)7^x$. Then, $\max_{i \in [k-1]} h(i) = h(k-1)$.*

Proof sketch. Differentiate to get $h'(x) = (k-x)7^x - x7^x + x(k-x)(\log 7)7^x = 7^x \cdot [x^2(-\log 7) + x(k \log 7 - 2) + k]$. It vanishes at $x = \left(\frac{k}{2} - \frac{1}{\log 7}\right) + \sqrt{\left(\frac{k}{2} - \frac{1}{\log 7}\right)^2 - \frac{k}{\log 7}}$. Thus, h is maximized at the integer $x = k-1$. ◀

C Details for Section 3.2

▶ **Definition 20** (Faithful hom.). *$\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{y}]$ is faithful for $\boldsymbol{T}$ if $\mathrm{trdeg}_{\mathbb{F}}(\boldsymbol{T}) = \mathrm{trdeg}_{\mathbb{F}}(\Phi(\boldsymbol{T}))$.*

The following fact about faithful maps is from [5, Thm. 2.4].

▶ **Fact 1** (Faithful is useful). *For any $C \in \mathbb{F}[y_1, \ldots, y_m]$, $C(\boldsymbol{T}) = 0 \iff C(\Phi(\boldsymbol{T})) = 0$.*

Here is an important criterion about the jacobian matrix which basically shows that it *preserves* algabraic independence.

▶ **Fact 2** (Jacobian criterion). *Let $\mathbf{f} \subset \mathbb{F}[\boldsymbol{x}]$ be a finite set of polynomials of degree at most d, and $\mathrm{trdeg}_{\mathbb{F}}(\mathbf{f}) \le r$. If char$(\mathbb{F}) = 0$, or char$(\mathbb{F}) > d^r$, then $\mathrm{trdeg}_{\mathbb{F}}(\mathbf{f}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{x})} \mathcal{J}_{\boldsymbol{x}}(\mathbf{f})$.*

The following lemma (& the proof) is similar to [5, Lem. 2.7]. It is a recipe to 'drastically' reduce variables, if trdeg is small.

▶ **Lemma 21** (Recipe for faithful maps). *Let $\boldsymbol{T} \in \mathbb{F}[\boldsymbol{x}]$ be be a finite set of polynomials of degree at most d and $\mathrm{trdeg}_{\mathbb{F}}(\boldsymbol{T}) \le r$, and char(F)=0 or $> d^r$. Let $\Psi' : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_1, z_2]$ such that $\mathrm{rk}_{\mathbb{F}(\boldsymbol{x})} \mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}) = \mathrm{rk}_{\mathbb{F}(\boldsymbol{z})} \Psi'(\mathcal{J}_{\boldsymbol{x}}(\boldsymbol{T}))$.*

Then, the map $\Phi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[\boldsymbol{z}, t, \boldsymbol{y}]$, such that $x_i \mapsto (\sum_j y_j t^{ij}) + \Psi'(x_i)$, is a faithful homomorphism for $\boldsymbol{T}$.

C.1 Technical Details for Theorem 2b

▷ Claim 22. Let $g_i \in L(T_i)$, where $T_i \in \Pi\Sigma\Pi^{[\delta]}$ of size atmost s, then $J_{\boldsymbol{x}_k}(g_1,\ldots,g_k) \in \Sigma^{[k!]}\Pi^{[k]}\Sigma\Pi^{[\delta-1]}$ of size $O(k!\,ks)$.

Proof sketch. Each entry of the matrix has degree at most $\delta - 1$. Trivial expansion gives $k!$ top-fanin where each product (of fanin k) has size $\sum_i \text{size}(g_i)$. As, $\text{size}(T_i) \le s$, trivially each $\text{size}(g_i) \le s$. Therefore, the total size is $k! \cdot \sum_i \text{size}(g_i) = O(k!\,ks)$. ◀

▷ Claim 23. Let $g \in \Sigma\Pi^{\delta}$, then $\Psi(g) \in \Sigma\Pi^{\delta}$ of size $3^{\delta} \cdot \text{size}(g)$ (for $n \gg \delta$).

Proof sketch. Each monomial $\boldsymbol{x}^{\boldsymbol{e}}$ of degree δ, can produce $\prod_i (e_i + 1) \le ((\sum_i e_i + n)/n)^n \le (\delta/n + 1)^n$-many monomials, by AM-GM inequality as $\sum_i e_i \le \delta$. As $\delta/n \to 0$, we have $(1+\delta/n)^n \to e^{\delta}$. As $e < 3$, the upper bound follows. ◀

[25, Prop. 4.18] gave the first nontrivial PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits:

▶ **Theorem 24** ([25]). *There is a* $\mathsf{poly}(n,d,\delta\log s)$*-explicit hitting set of size* $(nd)^{O(\delta \log s)}$ *for the class of* n*-variate, degree-*$(\le d)$ *polynomials* $f(\boldsymbol{x})$*, computed by* $\Sigma\wedge\Sigma\Pi^{[\delta]}$*-circuit of size* s*.*

D Proof sketch of Theorem 2a: Similar to Section 3.2

Similar to Theorem 2b, we generalize this theorem and prove for a much bigger class of polynomials.

▶ **Problem 2.** *Let* $\{T_i \,|\, i \in [m]\}$ *be* $\Pi\Sigma\wedge$ *circuits of (syntactic) degree at most* d *and size* s*. Let the transcendence degree of* T_i*'s,* $\text{trdeg}_{\mathbb{F}}(T_1,\ldots,T_m) =: k \ll s$*. Further,* $C(x_1,\ldots,x_m)$ *be a circuit of size + degree* $< s'$*. Design a blackbox-PIT algorithm for* $C(T_1,\ldots,T_m)$*.*

It is trivial to see that $\Sigma^{[k]}\Pi\Sigma\wedge$ is a very special case of the above settings. We will use the same idea (& notation) as in Theorem 2b, using the Jacobian technique. The main idea is to come up with Φ map, and correspondingly the hitting set H'. If $g \in L(T_i)$, then $\text{size}(g) \le O(dn)$. We also note that D_1, which is an upper bound on $\deg_{z_2}\Psi(g)$ is $\mathsf{poly}(n,d)$ (Lemma 10). The D (and hence $R[\boldsymbol{x}]$) remains as before. Claims 5-6 hold similarly. We will construct the hitting set H' by showing that F has a small $\Sigma\wedge\Sigma\wedge$ circuit over $R[\boldsymbol{x}]$.

Note that, Claim 22 remains the same for $\Sigma\wedge\Sigma\wedge$ (implying the same size blowup). However, Claim 23, the size blowup is $O(d\,\text{size}(g))$, because each monomial x^e can only produce $d+1$ many monomials. Therefore, similar to Lemma 9, one can show that $\Psi(J_{\boldsymbol{x}_k}(g_1,\ldots,g_k)) \in \Sigma\wedge\Sigma\wedge$, of size $O(2^k k! kds)$. Similarly, the size in Lemma 8 can be replaced by $s^{O(k)}$. Therefore, we get (similar to Claim 7):

▷ Claim 25. $F \in R[\boldsymbol{x}]$ has $\Sigma\wedge\Sigma\wedge$-circuit of size $s^{O(k)}$.

Next, the degree bound also remains the same (except the parameter D_1 which is now $\mathsf{poly}(nd)$). Following the same footsteps, it is not hard to see that the degree bound of z_2 on P and Q, where $F = P(\boldsymbol{x},\boldsymbol{z})/Q(z_2)$, is $s^{O(k)}\mathsf{poly}(nd)$, while degree bound on z_1 remains $\mathsf{poly}(ksd)$. Therefore, $P \in \mathbb{F}[\boldsymbol{x},\boldsymbol{z}]$ has $\Sigma\wedge\Sigma\wedge$-circuit of size $s^{O(k)}$.

We want to *construct* a set $H' \subseteq \mathbb{F}^n$ such that the action $P(H',\boldsymbol{z}) \ne 0$. By Theorem 18, we conclude that it has $s^{O(k\ \log\log s)}$ size hitting set which is constructible in a similar time. Hence, the construction of map Φ and the theorem follows (from $\boldsymbol{z}$-degree bound).

Solution to Problem 2. We know that $C(T_1,\ldots,T_m) = 0 \iff E := \Phi(C(T_1,\ldots,T_m)) = 0$. Since, H' can be constructed in $s^{O(k\log\log s)}$ time, it is trivial to find hitting set for $E|_{H'}$ (which is just a $k+3$-variate polynomial with the aforementioned degree bounds). The final hitting set for E can be constructed in $s'^{O(k)} \cdot s^{O(k\log\log s)}$ time. ◀

E Algorithm for Theorem 1

The whitebox PIT for Theorem 1, that is discussed in Section 3.1, appears (below) as Algorithm 1.

Algorithm 1 Whitebox PIT Algorithm for $\Sigma^{[k]}\Pi\Sigma\wedge$-circuits

Input : $f = T_1 + \ldots + T_k \in \Sigma^{[k]}\Pi\Sigma\wedge$, a whitebox circuit of size s over $\mathbb{F}[\boldsymbol{x}]$.
Output : 0, if $f \equiv 0$, and 1, if it is non-zero.

Let $\Psi : \mathbb{F}[\boldsymbol{x}] \longrightarrow \mathbb{F}[z_2]$, be a sparse-PIT map, using [49] (Theorem 10). Apply it on f and check whether $\Psi(f) \stackrel{?}{=} 0$. If non-zero, `output` 1 otherwise, apply $\Phi : x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$ on f. Check $\sum_{i\in[k-1]} \partial_{z_1}(\Phi(T_i)/\Phi(T_k)) \stackrel{?}{=} 0 \bmod z_1^{d_1}$ $(d_1 := s)$ as follows:

Consider each $T_{i,1} := \partial_{z_1}(\Phi(T_i)/\Phi(T_k))$ over $R_1(\boldsymbol{x})$, where $R_1 := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_1}\rangle$. Use dlog computation (Claim 4), to write each $T_{i,1}$ in a 'bloated' form as $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$.

for $j \leftarrow 1$ **to** $k-1$ **do**

Reduce the top-fanin at each step using '**Di**vide & **De**rive' technique. Assume that at j-th step, we have to check the identity:

$$\sum_{i\in[k-j]} T_{i,j} \stackrel{?}{=} 0 \text{ over } R_j(\boldsymbol{x}), \text{ where } R_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j}\rangle\,,$$

each $T_{i,j}$ has a $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ representation and therein each $\Pi\Sigma\wedge|_{z_1=0} \in \mathbb{F}(z_2)\setminus\{0\}$.

1. Compute $v_{k-j,j} := \min_i \mathrm{val}_{z_1}(T_{i,j})$; by reordering it is for $i = k-j$. To compute $v_{k-j,j}$, use coefficient extraction (Lemma 13) and $\Sigma\wedge\Sigma\wedge$-circuit PIT (Lemma 18).
2. '**Di**vide' by $T_{k-j,j}$ and check whether

$$\left.\left(\sum_{i\in[k-j-1]} \frac{T_{i,j}}{T_{k-j,j}} + 1\right)\right|_{z_1=0} \stackrel{?}{=} 0.$$

Note: this expression is in $(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. Use— (1) $\Pi\Sigma\wedge|_{z_1=0} \in \mathbb{F}(z_2)$, and (2) *closure* of $\Sigma\wedge\Sigma\wedge$ under multiplication. Finally, do PIT on this by Lemma 18.

3. If it is non-zero, `output` 1, otherwise '**De**rive' wrt z_1 and '**In**duct' on $\left(\sum_{i\in[k-j-1]} \partial_{z_1}(T_{i,j}/T_{k-j,j})\right) \stackrel{?}{=} 0$, over $R_{j+1}(\boldsymbol{x})$ where $R_{j+1} := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j - v_{k-j,j}-1}\rangle$.
4. Again using dlog (Claim 4), show that $T_{i,j+1} := \partial_{z_1}(T_{i,j}/T_{k-j,j})$ has small $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$-circuit over $R_{j+1}(\boldsymbol{x})$. So call the algorithm on $\sum_{i\in[k-j-1]} T_{i,j+1} \stackrel{?}{=} 0$.

$j \leftarrow j+1$.

end

At the end, $j = k-1$. Do PIT (Lemma 18) on the single $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ circuit, over $R_{k-1}(\boldsymbol{x})$. If it is zero, `output` 0 otherwise `output` 1.

Words of caution: Throughout the algorithm there are intermediate expressions to be stored compactly. Think of them as 'special' circuits in $\boldsymbol{x}$, but over the *function-field* $\mathbb{F}(\boldsymbol{z})$. Keep track of their degrees wrt z_1, z_2; and that of the sizes of their fractions represented in 'bloated' circuit form.