Hardness vs Randomness for Bounded Depth Arithmetic Circuits

Chi-Ning Chou

School of Engineering and Applied Sciences, Harvard University,
Cambridge, MA 02138, USA
chiningchou@g.harvard.edu

Mrinal Kumar

Center for Mathematical Sciences and Applications, Harvard University
Cambridge, MA 02138, USA
mrinalkumar08@gmail.com

Noam Solomon

Center for Mathematical Sciences and Applications, Harvard University
Cambridge, MA 02138, USA
noam.solom@gmail.com

Abstract

In this paper, we study the question of hardness-randomness tradeoffs for bounded depth arithmetic circuits. We show that if there is a family of explicit polynomials $\{f_n\}$, where f_n is of degree $O(\log^2 n/\log^2 \log n)$ in n variables such that f_n *cannot* be computed by a depth Δ arithmetic circuits of size $\mathsf{poly}(n)$, then there is a deterministic sub-exponential time algorithm for polynomial identity testing of arithmetic circuits of depth $\Delta - 5$.

This is incomparable to a beautiful result of Dvir et al.[SICOMP, 2009], where they showed that super-polynomial lower bounds for depth Δ circuits for any explicit family of polynomials (of potentially high degree) implies sub-exponential time deterministic PIT for depth $\Delta - 5$ circuits of *bounded individual degree*. Thus, we remove the "bounded individual degree" condition in the work of Dvir et al. at the cost of strengthening the hardness assumption to hold for polynomials of *low* degree.

The key technical ingredient of our proof is the following property of roots of polynomials computable by a bounded depth arithmetic circuit : if $f(x_1, x_2, \ldots, x_n)$ and $P(x_1, x_2, \ldots, x_n, y)$ are polynomials of degree d and r respectively, such that P can be computed by a circuit of size s and depth Δ and $P(x_1, x_2, \ldots, x_n, f) \equiv 0$, then, f can be computed by a circuit of size $\mathsf{poly}(n, s, r, d^{O(\sqrt{d})})$ and depth $\Delta + 3$. In comparison, Dvir et al. showed that f can be computed by a circuit of depth $\Delta + 3$ and size $\mathsf{poly}(n, s, r, d^t)$, where t is the degree of P in y. Thus, the size upper bound in the work of Dvir et al. is non-trivial when t is small but d could be large, whereas our size upper bound is non-trivial when d is small, but t could be large.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Algebraic Complexity, Polynomial Factorization Circuit Lower Bounds, Polynomial Identity Testing

Digital Object Identifier 10.4230/LIPIcs.CCC.2018.13

Related Version https://eccc.weizmann.ac.il/report/2018/052/

Acknowledgements We are thankful to Rafael Oliveira and Guy Moshkovitz for helpful discussions.

33rd Computational Complexity Conference (CCC 2018).
Editor: Rocco A. Servedio; Article No. 13; pp. 13:1–13:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Arithmetic circuits are one of the most natural and fundamental models of algebraic computation. Formally, an arithmetic circuit Ψ over a field $\mathbb{F}$ and variables $\vec{x} = (x_1, x_2, \ldots, x_n)$ is a directed acyclic graph, with the gates of in-degree zero (called leaves) being labeled by elements in $\mathbb{F}$ and variables in $\vec{x}$, and the internal nodes being labeled by $+$ (sum gates) or $\times$ (product gates). The vertices of out-degree zero in Ψ are called output gates. The circuit Ψ computes a polynomial in $\mathbb{F}[\vec{x}]$ in a natural way : the leaves compute the polynomial equal to its label. A sum gate computes the polynomial equal to the sum of the polynomials computed at its children, while a product gate computes the polynomial equal to the product of the polynomials computed at its children. Arithmetic circuits can be thought of as algebraic analog of Boolean circuits, and provide a succinct representation of multivariate polynomials, and are natural objects of study in Algebraic Complexity theory. Two of the most fundamental problems of interest in this area of research are the following.

- **Lower Bounds.** To show that there are *explicit* polynomial families which are hard, i.e. they cannot be computed by arithmetic circuits whose size is polynomial in the number of variables.
- **Polynomial Identity Testing (PIT).** To design an efficient deterministic algorithm which takes as input an arithmetic circuit C, and outputs if it is identically zero or not.

It is easy to show by an appropriate counting argument that a random polynomial of degree d in n variables cannot be computed by an arithmetic circuit of size $\mathsf{poly}(n, d)$, but no such *explicit*[1] polynomial families are known. Similarly, a randomized algorithm for the PIT question immediately follows from the classical Schwartz-Zippel lemma (see Lemma 15). The key challenge is to accomplish this task without using randomness.

The progress on these questions for general arithmetic circuits has been painfully slow. To date, there are no non-trivial[2] algorithms for PIT for general arithmetic circuits, while the best known lower bound, due to Bauer and Strassen [2], is a slightly superlinear lower bound $\Omega(n \log n)$, established over three decades ago. In fact, even for the class of bounded depth arithmetic circuits, no non-trivial deterministic PIT algorithms are known, and the best lower bounds known are just slightly superlinear [22].

In a very influential work, Kabanets and Impagliazzo [10] showed that the questions of derandomizing PIT and that of proving lower bounds for arithmetic circuits are equivalent in some sense. Their result adapts the Hardness vs Randomness framework of Nisan and Wigderson [18] to the algebraic setting. In their proof, Kabanets and Impagliazzo combine the use of Nisan-Wigderson generator with Kaltofen's result that all factors of a low degree (degree $\mathsf{poly}(n)$) polynomial with $\mathsf{poly}(n)$ sized circuit are computable by size $\mathsf{poly}(n)$ circuits [12]. They showed that given an explicit family of *hard* polynomials, one can obtain a *non-trivial*[3] deterministic algorithm for PIT.

The extremely slow progress on the lower bound and PIT questions for general circuits has led to a lot of attention on understanding these questions for more structured sub-classes of arithmetic circuits. Arithmetic formula [11], algebraic branching programs [15], multilinear circuits [21, 25, 24], and constant depth arithmetic circuits [19, 22, 9, 7, 17] are some examples of such circuit classes. A natural question is to ask if the equivalence of PIT and lower bounds

[1] See Definition 10 for a formal definition.

[2] Here, non-trivial means anything which is better than the brute force algorithm for general arithmetic circuits given by the Schwartz-Zippel lemma.

[3] Here, non-trivial means subexponential time, or quasipolynomial time, based on the hardness assumption.

also carries over to these more structured circuit classes. For example, does super-polynomial lower bounds for arithmetic formulas imply non-trivial deterministic algorithms for PIT for arithmetic formulas, and vice-versa?

The answers to these questions do not follow directly from the results in [10]; unlike general arithmetic circuits, none of these sub-classes are known to be *closed* under factoring, i.e., given a polynomial P which has a small formula (or bounded depth circuit), it is not known whether the factors of P also have small formulas (or bounded depth circuits). Recently, there has been some progress on these questions (see [20, 4]), but in general, these questions of being closed under factoring for arithmetic formulas and bounded depth circuits continue to remain open.

1.1 Bounded Depth Circuits

Dvir, Shpilka and Yehudayoff [5] initiated the study of this question of equivalence of PIT and lower bounds for bounded depth circuits. Dvir et al. observed that a part of the proof in [10] can be generalized to show that non-trivial PIT for bounded depth circuits implies lower bounds for such circuits. For the converse, the authors only showed a weaker statement; they proved that super-polynomial lower bounds for depth Δ arithmetic circuit implies non-trivial PIT for depth $\Delta - 5$ arithmetic circuits with *bounded individual degree*. The bounded individual degree condition is a bit unsatisfying, and so, the following question is of fundamental interest.

▶ **Question 1.** *Does a super-polynomial lower bound for depth Δ arithmetic circuits imply non-trivial deterministic PIT for depth Δ' arithmetic circuits*[4]*? In particular, can we get rid of the "bounded individual degree" condition from the results in [5]?*

In this paper, we partially answer Question 1 in the affirmative. Informally, we prove the following theorem.

▶ **Theorem 2** (Informal). *A super-polynomial lower bound for depth Δ arithmetic circuits for an explicit family of* low degree *polynomials implies non-trivial deterministic PIT for depth $\Delta - 5$ arithmetic circuits.*

Here, by low degree polynomials, we mean polynomials in n variables and degree at most $O(\log^2 n/ \log^2 \log n)$. Thus, by strengthening the hardness hypothesis in [5], we remove the bounded individual degree restriction from the implication. We now formally state our results and elaborate further how they compare with prior work.

1.2 Our Results

We start by stating our main theorem, which is a formal restatement of Theorem 2.

▶ **Theorem 3.** *Let $\Delta \geq 6$ be a positive integer, and let $\varepsilon > 0$ be any real number. Let $\{f_m\}$ be a family of explicit polynomials such that f_m is an m-variate multilinear polynomial of degree $d = O\left(\log^2 m/\log^2 \log m\right)$ which cannot be computed by an arithmetic circuit of depth Δ and size* $\mathsf{poly}(m)$*. Then, there is a deterministic algorithm, which, given as input a circuit $C \in \mathbb{C}[\vec{x}]$ of size s, depth $\Delta - 5$ and degree D on n variables, runs in time $(snD)^{O(n^{2\varepsilon})}$ and determines if the polynomial computed by C is identically zero.*

[4] Here, we think of Δ' as $\Delta - O(1)$.

Some remarks on the above theorem statement.

▶ Remark. Our algorithm works as long as the characteristic of the underlying field is sufficiently large or zero, but for simplicity, the presentation in this paper just focuses on the field $\mathbb{Q}$ of rational numbers.

▶ Remark. The bound $d \leq \log^2 m/\log^2 \log m$ can be relaxed to $d \leq \log^k m/\log^k \log m$ for any positive integer k, but we would need lower bounds for depth $\Delta + 2k + 2$ to be able to do PIT for depth Δ circuits. We point this difference out in the proof of Theorem 5, but do not dwell further on this.

▶ Remark. The running time of the PIT algorithm gets better as the lower bound gets stronger. Also, the constraint on the degree of the hard polynomial family can be further relaxed a bit, at the cost of strengthening the hardness assumption, and increasing the running time of the resulting PIT algorithm[5]. We leave it to the interested reader to work out these details.

▶ Remark. In general, explicit polynomial families do not have to be multilinear. But, if we have a hard polynomial which is not multilinear, and has a polynomial degree in each variable, we can derive from it an explicit hard multilinear polynomial with only a polynomial deterioration in the hardness parameters. More precisely, replacing x_i^r, for $r > 1$ with $y_{i_0}^{r_0} \cdot \ldots \cdot y_{i_s}^{r_s}$. where $(r_0 \ldots r_s)$ is the binary representation of r, gives a new multilinear polynomial in a slightly larger number of variables. This polynomial is at least as hard as the original polynomial which can be recovered from it by the substitution $y_{ij} = x_i^{2^j}$.

As discussed earlier, Theorem 3 is closely related to the main result in [5]. We now discuss their similarities and differences.

Comparison with [5]

- **Degree constraint on the hard polynomial.** While Theorem 3 requires that the hard polynomial on m variables has degree at most $O(\log^2 m/\log^2 \log m)$, Dvir et al. [5] did not have a similar constraint.
- **Individual degree constraint for PIT.** In [5], the authors get PIT for low depth circuits with bounded individual degree, whereas our Theorem 3 does not make any assumptions on individual degrees in this context.

As we alluded to earlier, the key technical challenge for extending the known hardness-randomness tradeoffs for general circuits [10] to restricted circuit classes like formulas or bounded depth circuits comes from the absence of an analog of Kaltofen's result [12] about closure under factoring for these restricted classes. More specifically, understanding the following questions seems necessary for adapting the proof strategy in [10] to other restricted classes of circuits.

▶ **Question 4.** *Let $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$ be a polynomial of degree r and let $f \in \mathbb{F}[\vec{x}]$ be a polynomial of degree d such that $P(\vec{x}, f) \equiv 0$. Assuming P can be computed by a low depth circuit (or arithmetic formula) of size at most s, can f be computed by a low depth circuit (or arithmetic formula) of size at most* $\mathsf{poly}(s, n, d, r)$*?*

In [5], the authors partially answer this question by showing that under the hypothesis of Question 4, the polynomial f can be computed by a low depth circuit of size at most

[5] If we assume a sub-exponential lower bound, then we can get a quasi-polynomial time PIT. Note that this is the parameter region used in [5]

$\mathsf{poly}(s, r, d^{\deg_y(P)})$. Thus, for the case of polynomials P which have small individual degree with respect to y, they answer the question in affirmative.

Our main technical observation is the following result, which gives an upper bound on the *low depth* circuit complexity of roots of low degree of a multivariate polynomial which has a small low depth circuit.

▶ **Theorem 5.** *Let $P \in \mathbb{F}[\vec{x}, y]$ be a polynomial of degree at most r in $n + 1$ variables that can be computed by an arithmetic circuit of size s of depth at most Δ. Let $f \in \mathbb{F}[\vec{x}]$ be a polynomial of degree at most d such that*

$$P(\vec{x}, f) = 0\,.$$

Then, f can be computed by a circuit of depth at most $\Delta+3$ and size at most $O((srn)^{10} d^{O(\sqrt{d})})$.

1.3 Proof Overview

The proof of Theorem 3 is very much along the lines of the proofs of similar results in [10] and [5]. In particular, all our technical contributions are confined to the proof of Theorem 5, which when combined with the standard machinery of Nisan-Wigderson designs yields Theorem 3. Our proof of Theorem 5 also mirrors the proof of the analogous theorem about the structure of roots in [5]. We now outline the main steps, and point out the differences between the proofs.

The first step in the proof is to show that one can use the standard Hensel Lifting to iteratively obtain better approximations of the root f given a circuit for $P(\vec{x}, y)$. More formally, in the k^{th} step, we start with a polynomial h_k which agrees with f on all monomials of degree at most k, and use it to obtain a polynomial h_{k+1} which agrees with f on all monomials of degree at most $k + 1$. Moreover, the proof shows that if h_k has a small circuit, then h_{k+1} has a circuit which is only slightly larger than that of h_k. This iterative process starts with the constant term of f, which trivially has a small circuit. Thus, after d iterations, we have a polynomial h_d such that the root f is the sum of the homogeneous components of h_d of degree at most d. This lifting step is exactly the same as that in [5] or in some of the earlier works on polynomial factorization [3], and is formally stated in Lemma 16.

The key insight of Dvir et al. [5] was that if $\deg_y(P) = t$, and $C_0(\vec{x}), C_1(\vec{x}), \ldots, C_t(\vec{x})$ are polynomials such that $P(\vec{x}, y) = \sum_{i=1}^{t} C_i(\vec{x}) y^t$, then for every $k \in \{0, 1, \ldots, d\}$, we have a polynomial B_k of degree at most k such that

$$h_k(\vec{x}) = B_k(C_0(\vec{x}), C_1(\vec{x}), \ldots, C_t(\vec{x}))\,.$$

Now, consider the case when $t \ll n$ (for instance $t = O(1)$). It follows from standard interpolation results for low depth circuits (see Lemma 12) that each of the polynomials $C_i(\vec{x})$ has a circuit of size $O(sr)$ and depth Δ since P has a polynomial of size s and depth Δ. Thus, $h_d(\vec{x})$ can be written as a sum of at most $\binom{d+t}{t} = O(d^t)$ monomials if we treat each C_i as a formal variable. Plugging in the small depth Δ circuits for each C_i, and standard interpolation (Lemma 12), it follows that f has a circuit of size $\mathsf{poly}(s, n, d^t)$ of depth $\Delta + O(1)$.

Observe that this size bound of $\mathsf{poly}(s, n, d^t)$ is small only when t is small. For instance, when $t > n$, this bound becomes trivial. Our key observation is that independently of t, there is a set of $d + 1$ polynomials $g_0(\vec{x}), g_1(\vec{x}), \ldots, g_d(\vec{x})$ of degree at most d, and polynomials $A_0, A_1, \ldots, A_k$ on $d + 1$ variables such that for every $k \in \{0, 1, \ldots, d\}$,

$$h_k(\vec{x}) = A_k(g_0(\vec{x}), g_1(\vec{x}), \ldots, g_d(\vec{x}))\,.$$

Moreover, for every k, A_k has degree at most k and is computable by a circuit of size at most $O(d^3)$. This observation essentially decouples the number of *generators* from the individual degree of P in y, and is formally stated as Lemma 18. Also, each of these generators g_i can be computed by a circuit of size $\mathsf{poly}(s, r)$ and depth Δ. Thus, expressing $A_d(z_0, z_1, \ldots, z_d)$ as a sum of monomials, and then composing this representation with the circuits for $g_0, g_1, \ldots, g_d$ would give us a circuit of size $\mathsf{poly}(s, n, r, d, 4^d)$ of depth $\Delta + O(1)$. To get a sub-exponential dependence on d in the size, we do not write $A_d(z_0, z_1, \ldots, z_d)$ as $\sum\prod$ circuit of size $O(4^d)$, but instead express it as a $\sum\prod\sum$ circuit of size at most $d^{O(\sqrt{d})}$, using the depth reduction result of [8][6].

One point to note is that just from Kaltofen's result [12], it follows that f has an arithmetic circuit[7] of size $\mathsf{poly}(n)$. Thus, from Theorem 9, it follows that f has a circuit of depth-3 of size at most $n^{O(\sqrt{d})}$. The key advantage of Theorem 5 over this bound is that the exponential term is $d^{O(\sqrt{d})}$ and not of the form $n^{d^{\varepsilon}}$. For $d \leq \log^2 n / \log^2 \log n$, $d^{O(\sqrt{d})}$ is bounded by a polynomial in n and so the final bound is meaningful.

We end this section with a short discussion on the *low degree* condition in the hypothesis of Theorem 3.

1.4 The Low Degree Condition

An intriguing question is to understand how restrictive the "low degree" condition in the hardness assumption of Theorem 3 is. More formally, is the question of proving super-polynomial lower bounds for constant depth circuits for an explicit polynomial family of low degree much harder than the question of proving super-polynomial lower bound for constant depth circuits for an explicit polynomial family of potentially larger degree [8]? Currently, we do not even know quadratic lower bounds for arithmetic circuits of constant depth, and so, perhaps we are quite far from understanding this question.

It is, however, easy to see that some of the known lower bounds for low depth circuits carries over to the low degree regime. For instance, the proofs of super-polynomial lower bounds for homogeneous depth-3 circuits by Nisan and Wigderson [19], super-polynomial lower bounds for homogeneous depth-4 circuits based on the idea of shifted partial derivatives (see for example, [9, 13, 7, 17]) and super-linear lower bound due to Raz [22] do not require the degree of the hard function to be large.

There are some known exceptions to this. For instance, lower bounds for homogeneous depth-5 circuits over finite fields due to Kumar and Saptharishi [16] are of the form $2^{\Omega(\sqrt{d})}$ and become trivial if $d < \log^2 n$. Another result which distinguishes the low degree and high degree regime is a separation between homogeneous depth-5 and homogeneous depth-4 circuit [16] which is only known to be true in the low degree regime (degree less than $\log^2 n$).

Another result of relevance is a result of Raz [23], which shows that constructing an explicit family of tensors $T_n : [n]^d \to \mathbb{F}$, of rank at least $n^{d(1-o(1))}$ implies super-polynomial lower bound for arithmetic formulas, provided $d \leq O(\log n / \log\log n)$. As far as we know, we do not know of such connections in the regime of high degree.

One prominent family of lower bound results which do not seem to generalize to this low degree regime are the super-polynomial lower bounds for multilinear formulas [21], and multilinear constant depth circuits [25]. In fact, the results in [23] show that super-polynomial

[6] See Theorem 9 for a formal statement of this result.
[7] Of potentially very large depth.
[8] In general, the degree only has to be upper bounded by a polynomial function in the number of variables.

lower bounds for set multilinear formulas for polynomials of degree at most $O(\log n/ \log \log n)$ implies super-polynomial lower bounds for general arithmetic formulas.

In the context of polynomial factorization, low degree factors of polynomials with small circuits have been considered before. For instance, Forbes [6] gave a quasi-polynomial time deterministic algorithm to test if a given polynomial of constant degree divides a given sparse polynomial. Extending this result to even testing if a given sparse polynomial divides another given sparse polynomial remains an open problem.

2 Preliminaries

We start by setting up some notation and stating some basic definitions and results from prior work which will be used in our proofs.

2.1 Notations

- We use boldface letters $\vec{x}, \vec{y}, \vec{z}$ to denote tuples of variables.
- For a polynomial P, $\deg(P)$ denotes the total degree of P and $\deg_y(P)$ denotes the total degree of P with respect to the variable y.
- Throughout this paper, we state and prove our results when the underlying field $\mathbb{F}$ is the field of rational numbers $\mathbb{Q}$, even though all our results hold as long as the field is of sufficiently large or zero characteristic.
- Let $P \in \mathbb{F}[\vec{x}]$ be a polynomial of degree equal to d. For every $k \in \mathbb{N}$, $\mathcal{H}_k\left[P\right]$ denotes the homogeneous component of P of degree k. Similarly, $\mathcal{H}_{\leq k}\left[P\right]$ is defined to be equal $\sum_{i=0}^{k} \mathcal{H}_i[P]$.
- For an arithmetic circuit C, we use $\text{size}(C)$ to denote the number of wires in C. The depth of C is the length of the longest path from any output gate to any input gate.
- Throughout this paper, we assume that all our circuits are layered with alternating layers of addition and multiplication gates. Moreover, we always assume that the top layer is of addition gates. For instance, a depth-3 circuit is of the form $\sum\prod\sum$ and a depth-4 circuit is of the form $\sum\prod\sum\prod$.

2.2 Derivatives

We start by defining derivatives of a polynomial. For the ease of presentation, we work with the notion of the slightly non-standard notion of *Hasse* derivatives even though we work with fields of characteristic zero.

▶ **Definition 6** (Derivatives). Let $\mathbb{F}$ be any field and let $P(y) \in \mathbb{F}[y]$ be a polynomial. Then for every $k \in \mathbb{N}$, the partial derivative of P of order k with respect to y denoted by $\frac{\partial^k P(y)}{\partial y^k}$ or $P^{(k)}(y)$ is defined as the coefficient of z^k in the polynomial $P(y+z)$.

We also use $P'(y)$ and $P''(y)$ to denote the first and second order derivatives of P respectively. An immediate consequence of this definition is the following lemma.

▶ **Lemma 7** (Taylor's expansion). *Let $P(y) \in \mathbb{F}[y]$ be a polynomial of degree d. Then,*

$$P(y+z) = P(y) + z \cdot P'(y) + z^2 \cdot P^{(2)}(y) + \cdots + z^d \cdot P^{(d)}(y)\,.$$

2.3 Depth Reductions

We will use the following depth reduction theorems as a blackbox for our proofs.

▶ **Theorem 8** (Depth reduction to depth-$2k$ [1, 14, 28]). *Let k be a positive integer and $\mathbb{F}$ be any field. If $P(\vec{x}) \in \mathbb{F}[\vec{x}]$ is an n-variate polynomial of degree d that be computed by an arithmetic circuit Ψ of size at most s, then P can be computed by a depth $2k$ circuit of size at most $(snd)^{O(d^{1/k})}$.*

Invoked with $k = 2$ the above theorem gives a circuit of depth 4 for the polynomial P of size $s^{O(\sqrt{d})}$. The next depth reduction result gives a further reduction to depth-3, as long as the field is of characteristic zero, and will be useful for our proof.

▶ **Theorem 9** (Depth reduction to depth-3 [8]). *Let $P(\vec{x}) \in \mathbb{Q}[\vec{x}]$ be an n-variate polynomial of degree d that can be computed by an arithmetic circuit Ψ of size at most s. Then, P can be computed by a $\sum\prod\sum$ circuit of size at most $(snd)^{O(\sqrt{d})}$.*

2.4 Explicit Polynomials

▶ **Definition 10** ([5]). Let $\{f_m\}$ be a family of multilinear polynomials such that $f_m \in \mathbb{F}[x_1, x_2, \ldots, x_m]$ for every m. Then, the family $\{f_m\}$ is said to be explicit if the following two conditions hold.

- All the coefficients of f_m have bit complexity polynomial in m.
- There is an algorithm which on input m outputs the list of all 2^m coeffcients of f_m in time $2^{O(m)}$.

2.5 Extracting Homogeneous Components

For our proofs, we will also rely on the following classical result of Strassen, which shows that if a polynomial P has a small circuit, then all its low degree homogeneous components also have small circuits.

▶ **Theorem 11** (Homogenization). *Let $\mathbb{F}$ be any field, and let $\Psi \in \mathbb{F}[\vec{x}]$ be an arithmetic circuit of size at most s. Then, for every $k \in \mathbb{N}$, there is a homogeneous circuit Ψ_k of formal degree at most k and size at most $O(k^2 s)$, such that*

$$\Psi_k = \mathcal{H}_k [\Psi] .$$

Theorem 11 gives us a way of extracting homogeneous components of the polynomial computed by a given circuit. One drawback of Theorem 11 is that the depth of Ψ_k could be much larger than the depth of Ψ. Thus, given a low depth circuit (and hence, unbounded in-degree circuit) for a polynomial P, it is not clear if the homogeneous components of P also have small low depth circuits. The following standard trick implies this observation, and would be useful for our proof.

▶ **Lemma 12** (Interpolation). *Let $\mathbb{F}$ be any field with at least $d + 1$ elements. Let $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$ be a polynomial of degree at most d. Let $C_0(\vec{x}), C_1(\vec{x}), \ldots, C_d(\vec{x}) \in \mathbb{F}[\vec{x}]$ be polynomials such that $P(\vec{x}, y) = \sum_{j=0}^{d} y^j \cdot C_j(\vec{x})$. Then, if $P(\vec{x}, y)$ has a circuit of size at most s and depth at most Δ, then for every $j \in \{0, 1, \ldots, d\}$, $C_j(\vec{x})$ has a circuit of size at most $O(sd)$ and depth Δ.*

We refer the reader to excellent surveys of Shpilka and Yehudayoff [27] and Saptharishi [26] for a proof of these results.

2.6 Hitting Sets

▶ **Definition 13.** A set of points $\mathcal{P}$ is said to be a hitting set for a class $\mathcal{C}$ of circuits, if for every $C \in \mathcal{C}$ which is not identically zero, there is an $\vec{a} \in \mathcal{P}$ such that $C(\vec{a}) \neq 0$.

Clearly, deterministic and efficient construction of a hitting set of small size for a class $\mathcal{C}$ of circuits immediately implies a deterministic PIT algorithm for $\mathcal{C}$. PIT algorithms designed in this way are also *blackbox*, in the sense that they do not have to look inside into the wiring of the circuit to decide if it computes a polynomial which is identically zero. The PIT algorithms in this paper are all blackbox in this sense.

2.7 Nisan-Wigderson Designs

We state the following well known result of Nisan and Wigderson [18] on the explicit construction of combinatorial designs.

▶ **Theorem 14** ([18]). *Let n, m be positive integers such that $n < 2^m$. Then, there is a family of subsets $S_1, S_2, \ldots, S_n \subseteq [\ell]$ with the following properties.*

- *For each $i \in [n]$, $|S_i| = m$.*
- *For each $i, j \in [n]$, such that $i \neq j$, $|S_i \cap S_j| \leq \log n$.*
- *$\ell = O(\frac{m^2}{\log n})$.*

Moreover, such a family of sets can be constructed via a deterministic algorithm in time $\mathsf{poly}(n, 2^\ell)$.

2.8 Schwartz-Zippel Lemma

We now state the well known Schwartz-Zippel lemma.

▶ **Lemma 15** (Schwartz-Zippel). *Let $\mathbb{F}$ be a field, and let $P \in \mathbb{F}[\vec{x}]$ be a non-zero polynomial of degree (at most) d in n variables. Then, for any finite set $S \subset \mathbb{F}$ we have*

$$|\{\vec{a} \in S^n : P(\vec{a}) = 0\}| \leq d|S|^{n-1}.$$

In particular, if $|S| \geq d+1$, then there exists some $\vec{a} \in S^n$ satisfying $P(\vec{a}) \neq 0$. This gives us a brute force deterministic algorithm, running in time $(d+1)^n$, to test if an arithmetic circuit computing a polynomial of degree at most d in n variables is identically zero.

3 Low Degree Roots of Polynomials with Shallow Circuits

In this section, we prove Theorem 5, which is also our main technical result. We start with the following lemma, which gives us a way of *approximating* the root of a polynomial to higher and higher accuracy, in an iterative manner. The lemma is a standard example of Hensel Lifting (in fact, sloppy Hensel Lifting), which appears in many of prior works in this area including [5]. The statement and the proof below, are from the work of Dvir et al [5].

▶ **Lemma 16** (Hensel Lifting [5]). *Let $P \in \mathbb{F}[\vec{x}, y]$ and $f \in \mathbb{F}[\vec{x}]$ be polynomials such that $P(\vec{x}, f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\vec{x}, f(\vec{x}))\right] = \delta \neq 0$. Let $i \in \{1, 2, \ldots, \deg(f)\}$ be any number. If $h \in \mathbb{F}[\vec{x}]$ is a polynomial such that $\mathcal{H}_{\leq i-1}[f] = \mathcal{H}_{\leq i-1}[h]$, then*

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h - \frac{P(\vec{x}, h)}{\delta}\right].$$

Proof. For the rest of the proof, we think of $P(\vec{x}, y)$ as an element of $\mathbb{F}[\vec{x}][y]$. Henceforth, we drop the variables $\vec{x}$ everywhere, and think of P as a univariate in y. Thus, $P(y) = P(\vec{x}, y)$. For brevity, we denote $\mathcal{H}_j[f]$ by f_j for every $j \in \mathbb{N}$.

From the hypothesis, we know that $P(f) = 0$. Therefore, $\mathcal{H}_{\leq i}(P(f)) = \mathcal{H}_{\leq i-1}\left[P(f)\right] = 0$. Moreover, since $\mathcal{H}_{\leq i-1}[h] = \mathcal{H}_{\leq i-1}[f]$, we get that $\mathcal{H}_{\leq i-1}\left[P(f)\right] = \mathcal{H}_{\leq i-1}\left[P(h)\right] = 0$. So, we have

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}\left[P(f)\right] \\ &= \mathcal{H}_{\leq i}\left[P\left(h + (f_i - h_i)\right)\right] \end{aligned}$$

Now, by using Lemma 7, we get the following equality.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}\left[P(h) + P'(h) \cdot (f_i - h_i) + P''(h) \cdot (f_i - h_1)^2 + \ldots + P^{(r)}(h) \cdot (f_i - h_1)^r\right] \\ &= \mathcal{H}_{\leq i}\left[P(h)\right] + \mathcal{H}_{\leq i}\left[P'(h) \cdot (f_i - h_i)\right] + \ldots + \mathcal{H}_{\leq i}\left[P^{(r)}(h) \cdot (f_i - h_i)^r\right] \end{aligned}$$

Here, r denotes the degree of P. Since every monomial in $f_i - h_i$ has degree equal to i, any term in the above summand which is divisible by $(f_i - h_i)^2$ does not contribute any monomial of degree at most i. Thus, we have the following.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}\left[P(h)\right] + \mathcal{H}_{\leq i}\left[P'(h) \cdot (f_i - h_i)\right] \\ &= \mathcal{H}_{\leq i}\left[P(h)\right] + \mathcal{H}_0\left[P'(h)\right] \cdot (f_i - h_i)\,. \end{aligned}$$

Now, we know that $\mathcal{H}_0\left[P'(h))\right] = \mathcal{H}_0\left[P'(f)\right] = \delta \neq 0$. Thus,

$$f_i = h_i - \frac{\mathcal{H}_i\left[P(h)\right]}{\delta}\,.$$

Since $\mathcal{H}_{\leq i-1}[P(h)]$ is identically zero, we get,

$$\mathcal{H}_{\leq i}\left[f\right] = \mathcal{H}_{\leq i}\left[h - \frac{P(h)}{\delta}\right]\,. \qquad ◀$$

For our proof, we shall look at the structure of the outcome of the lifting operation in Lemma 16 more closely. Before proceeding further, we need the following crucial lemma.

▶ **Lemma 17.** *Let $P(\vec{x}, y) \in \mathbb{F}[\vec{x}, y]$ be a polynomial of degree at most r, let $\alpha \in \mathbb{F}$ be a field element and $d \in \mathbb{N}$ be a positive integer. Let $\mathcal{G}'(P, \alpha, d)$ be the set of polynomials defined as follows.*

$$\mathcal{G}'(P, \alpha, d) = \left\{\mathcal{H}_{\leq d}\left[\frac{\partial^j P}{\partial y^j}\left(\vec{x}, \alpha\right)\right] - \mathcal{H}_0\left[\frac{\partial^j P}{\partial y^j}\left(\vec{x}, \alpha\right)\right] : j \in \{0, 1, 2, \ldots, d\}\right\}.$$

Let $\mathcal{G}(P, \alpha, d)$ be the subset of $\mathcal{G}'(P, \alpha, d)$ consisting of all non-zero polynomials. Then, the following statements are true.

- *For every $g \in \mathcal{G}(P, \alpha, d)$, the degree of every non-zero monomial in g is at least 1 and at most d.*
- *$|\mathcal{G}| \leq d + 1$.*
- *If P has a circuit of size at most s and depth Δ, then every $g \in \mathcal{G}(P, \alpha, d)$ has a circuit of size at most $O(sr^3d^2)$ and depth Δ.*

▶ Remark. $\mathcal{G}'$ contains the non-constant part of the partial derivatives of P at α up to order d. Note that $\mathcal{G}'$ may contain the zero polynomial, but $\mathcal{G}$ is the subset of $\mathcal{G}'$ without the zero polynomial.

Proof. The first two items follow immediately from the definition of $\mathcal{G}(P, \alpha, d)$. We focus on the proof of the third item. Let $C_0(\vec{x}), C_1(\vec{x}), \ldots, C_r(\vec{x})$ be polynomials such that

$$P(\vec{x}, y) = \sum_{i=0}^{r} C_i(\vec{x}) \cdot y^i \,.$$

Now, for any $j \in \{0, 1, 2, \ldots, d\}$, by Definition 6, $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$ is the coefficient of z^j in $P(\vec{x}, y+z)$. Moreover,

$$\begin{aligned} P(\vec{x}, y+z) &= \sum_{i=0}^{r} C_i(\vec{x}) \cdot (y+z)^i \,, \\ &= \sum_{i=0}^{r} C_i(\vec{x}) \cdot \left(\sum_{j=0}^{i} \binom{i}{j} z^j y^{i-j} \right) \,, \\ &= \sum_{j=0}^{r} \left(\sum_{i=j}^{r} \binom{i}{j} C_i(\vec{x}) \cdot y^{i-j} \right) \cdot z^j \,. \end{aligned}$$

Thus, for every $j \in \{0, 1, \ldots, d\}$, the coefficient of z^j in $P(\vec{x}, y+z)$ is given by $\sum_{i=j}^{r} \binom{i}{j} C_i(\vec{x}) \cdot y^{i-j}$. From Lemma 12, we know that each $C_i(\vec{x})$ has a circuit of depth Δ and size at most $O(sr)$. Thus, we can obtain a circuit for $\binom{i}{j} C_i(\vec{x}) \cdot y^{i-j}$ by adding an additional layer of × gates on top of the circuit for $C_i(\vec{x})$. This increases the size by a multiplicative factor of r, and the depth by 1. However, observe that this increase in depth is not necessary. Since, an expression of the form $y^i \cdot (\sum_a \prod_b Q_{a,b})$ can be simplified to $\sum_a y^i \cdot (\prod_b Q_{a,b})$. Thus, the multiplication by y^i can be absorbed in the product layer below the topmost layer of the circuits for $C_i(\vec{x})$, and this does not incur any additional increase in size. Thus, the polynomials $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$, and hence $\frac{\partial^j P}{\partial y^j}(\vec{x}, \alpha)$ have a circuit of size at most $O(sr^3)$ and depth at most Δ. To compute the homogeneous components of these polynomials, which are essentially the elements of $\mathcal{G}(P, \alpha, d)$, we just use Lemma 12. This increases the size by a factor of at most $O(d^2)$ while keeping the depth the same. ◀

We now state our key technical observation.

▶ **Lemma 18.** *Let $P \in \mathbb{F}[\vec{x}, y]$ and $f \in \mathbb{F}[\vec{x}]$ be polynomials of degree r and d respectively such that $P(\vec{x}, f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\vec{x}, f(\vec{x}))\right] = \delta \neq 0$. Let the polynomials in the set $\mathcal{G}(P, \mathcal{H}_0[f], d)$ be denoted by $g_0, g_1, \ldots, g_d$. Then, for every $i \in \{1, 2, \ldots, d\}$, there is a polynomial $A_i(\vec{z})$ in $d+1$ variables such that the following are true.*

- $\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(g_0, g_1, \ldots, g_d)]$, *and*
- $A_i(\vec{z})$ *is computable by a circuit of size at most $10d^2 i$.*

This is the analog of the main technical lemma in [5], which we state below.

▶ **Lemma 19 ([5]).** *Let $P \in \mathbb{F}[\vec{x}, y]$ and $f \in \mathbb{F}[\vec{x}]$ be polynomials of degree r and d respectively such that $P(\vec{x}, f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\vec{x}, f(\vec{x}))\right] = \delta \neq 0$. Let $P(\vec{x}, y) = \sum_{i=0}^{k} C_i(\vec{x}) \cdot y^i$. Then, for every $i \in \{1, 2, \ldots, \deg(f)\}$, there is a polynomial $A_i(\vec{z})$ in $k+1$ variables such that,*

$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}[A_i(C_0, C_1, \ldots, C_k)] \,.$$

The difference between these lemmas is that in [5], it is shown that there is a set of polynomials of size at most $\deg_y(P)+1$ which *generate* every homogeneous component of the root f. Thus, in the regime of bounded individual degree, the size of this generating set is very small. However, when $\deg_y(P) \geq n$, Lemma 19 does not say anything non-trivial since f can be trivially written as a polynomial in the n original variables. In contrast, Lemma 18 continues to say something non-trivial, as long as $d << n$, regardless of the value of $\deg_y(P)$. We now proceed with the proof.

Proof of Lemma 18. For the rest of the proof, we think of $P(\vec{x}, y)$ as an element of $\mathbb{F}[\vec{x}][y]$. So, we drop the variables $\vec{x}$ everywhere, and think of P as a univariate in y. Thus, $P(y) = P(\vec{x}, y)$. For brevity, we denote $\mathcal{H}_j[f]$ by f_j for every $j \in \mathbb{N}$. We also use $\mathcal{G}$ for $\mathcal{G}(P, f_0, d)$. The proof will be by induction on i and crucially use Lemma 16.

- **Base case.** We first prove the lemma for $i = 1$. We invoke Lemma 16 with $i = 1$ and $h = f_0$. We get that

$$\mathcal{H}_{\leq 1}[f] = \mathcal{H}_{\leq 1}\left[f_0 - \frac{P(f_0)}{\delta}\right] .$$

The proof follows by observing that f_0, δ are constants and $\mathcal{H}_1\left[P(f_0)\right] = \mathcal{H}_1\left[g_0\right]$ where $g_0 = \mathcal{H}_{\leq d}\left[P(f_0)\right] - \mathcal{H}_0\left[P(f_0)\right] \in \mathcal{G}$.

- **Induction step.** We assume that the claim in the lemma holds up to homogeneous components of degree at most $i-1$, and argue that it holds for $\mathcal{H}_{\leq i}[f]$. We invoke Lemma 16 with $h = A_{i-1}(g_0, g_1, \ldots, g_d)$, which exists by the induction hypothesis.

$$\mathcal{H}_{\leq i}\left[f\right] = \mathcal{H}_{\leq i}\left[h - \frac{P(h)}{\delta}\right] .$$

Recall that $\mathcal{H}_0(h) = \mathcal{H}_0(f)$. Thus, $h = f_0 + \tilde{h}$, where every monomial in $\tilde{h}$ has degree at least 1. By Lemma 7,

$$P(f_0 + \tilde{h}) = P(f_0) + P'(f_0) \cdot \tilde{h} + \cdots + P^{(r)}(f_0) \cdot \tilde{h}^r .$$

Thus, as $\tilde{h}$ has degree at least 1, we have

$$\begin{aligned}\mathcal{H}_{\leq i}\left[f\right] &= \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(P(f_0) + P'(f_0) \cdot \tilde{h} + \cdots + P^{(r)}(f_0) \cdot \tilde{h}^r\right)\right] ,\\ &= \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(P(f_0) + P'(f_0) \cdot \tilde{h} + \cdots + P^{(i)}(f_0) \cdot \tilde{h}^i\right)\right] .\end{aligned}$$

Since we are only interested in $i \leq d$, the following equality is also true.

$$\begin{aligned}&\mathcal{H}_{\leq i}\left[f\right]\\ &\quad = \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(\mathcal{H}_{\leq d}\left[P(f_0)\right] + \mathcal{H}_{\leq d}\left[P'(f_0)\right] \cdot \tilde{h} + \cdots + \mathcal{H}_{\leq d}\left[P^{(i)}(f_0)\right] \cdot \tilde{h}^i\right)\right] .\end{aligned}$$

Observe that for every $j \in \{0, 1, \ldots, d\}$, $\mathcal{H}_{\leq d}\left[P^{(j)}(f_0)\right]$ is an affine form in the elements of $\mathcal{G}$[9]. For every $j \in \{0, 1, 2, \ldots, i\}$, let $\ell_j(\vec{z})$ be an affine form such that $\ell_j(g_0, g_1, \ldots, g_d) =$

[9] In fact, they are an affine form in one variable.

$\mathcal{H}_{\leq d}\left[P^{(j)}(f_0)\right]$. Now, we define $A_i(\vec{z})$ as

$$A_i(\vec{z}) \equiv A_{i-1}(\vec{z}) - \frac{1}{\delta}\left(\ell_0(\vec{z}) + \ell_1(\vec{z}) \cdot (A_{i-1}(\vec{z}) - f_0) + \cdots + \ell_i(\vec{z}) \cdot (A_{i-1}(\vec{z}) - f_0)^i\right) .$$

The first item in the statement of the lemma is true, just by the definition of $A_i(\vec{z})$ above. We now argue about the circuit size of $A_i(\vec{z})$. Each affine form $\ell_i(\vec{z})$ can be computed by a circuit of size at most $O(d)$. Thus, given a circuit of $A_{i-1}(\vec{z})$, we can obtain a circuit for $A_i(\vec{z})$ by adding at most $10d^2$ additional gates. Thus, $A_i(\vec{z})$ can be computed by a circuit of size at most $10d^2(i-1) + 10d^2 = 10d^2 i$ gates. ◀

We are now ready to complete the proof of Theorem 5.

Proof of Theorem 5. The first step is to massage the circuit for P so that the hypothesis of Lemma 18 holds. We will have to keep track of the size and depth blow ups incurred in the process. We begin by ensuring that f is a root of multiplicity 1 of some polynomial related to P.

Reducing multiplicity of the root f

Let $P(\vec{x}, y) = \sum_{i=0}^{r} y^i C_i(\vec{x})$. Let $m \geq 1$ be the multiplicity of f as a root of $P(\vec{x}, y)$. Thus, $\frac{\partial^j P}{\partial y^j}(\vec{x}, f) = 0$ for $j \in \{0, 1, 2, \ldots, m-1\}$, but $\frac{\partial^m P}{\partial y^m}(\vec{x}, f) \neq 0$. The idea is to just work with the polynomial $\tilde{P} = \frac{\partial^{m-1} P}{\partial y^{m-1}}(\vec{x}, y)$ for the rest of the proof. Clearly, f is a root of multiplicity exactly 1 of $\tilde{P}$. We only need to ensure that $\tilde{P}$ can also be computed by a small low depth circuit. This follows from the proof of the third item in Lemma 17, where we argued that $\frac{\partial^j P}{\partial y^j}(\vec{x}, y)$ has a depth Δ circuit of size $O(sr^3)$.

Translating the origin

From the step above, we can assume without loss of generality that $\frac{\partial P}{\partial y}(\vec{x}, f) \neq 0$. Thus, there is a point $\vec{a} \in \mathbb{F}^n$ such that $\frac{\partial P}{\partial y}(\vec{a}, f(\vec{a})) \neq 0$. By translating the origin, we will assume that $\frac{\partial P}{\partial y}(0, f(0)) \neq 0$. This increases the depth of the circuit by at most 1, as it could involve replacing every variable x_i by $x_i + a_i$, and the size by at most a factor n.

Degree of A_d

From Lemma 18, we know that the polynomial $A_d(\vec{z})$ has a circuit of size at most $O(d^3)$. To obtain a circuit for f, we first prune away all the homogeneous components of $A_d(\vec{z})$ of degree larger than d. Recall that by definition, every polynomial $g_i \in \mathcal{G}$ has degree at least 1, and that $f = \mathcal{H}_{\leq d}[A_d(g_1, g_2, \ldots, g_d)]$. Thus, any monomial of degree strictly greater than d in $A_d(\vec{z})$ contributes no monomial of degree at most d in the variables $\vec{x}$ in the composed polynomial $A_d(g_1, g_2, \ldots, g_d)$, and hence does not contribute anything to the computation of f. So, we can confine ourselves to working with the homogeneous components of $A_d(\vec{z})$ of degree at most d.

By Theorem 11, we know that given a circuit for $A_d(\vec{z})$, we can construct a circuit for $\mathcal{H}_i[A_d(\vec{z})]$ by increasing the size of the circuit by a multiplicative factor of at most $O(i^2)$. Thus, $\mathcal{H}_{\leq d}[A_d(\vec{z})]$ can be computed by a circuit of size at most $O(d^3) \times \mathrm{size}(A_d(\vec{z}))$. Thus, for the rest of this argument, we will assume that $A_d(\vec{z})$ has a circuit of size at most $O(d^6)$ and degree at most d, and

$$f = \mathcal{H}_{\leq d}[A_d(g_1, g_2, \ldots, g_d)] .$$

Circuit for $A_d(\vec{z})$ of small depth

Given that $A_d(\vec{z})$ has a circuit of size $O(d^6)$ and degree at most d, by Theorem 9, we know that $A_d(\vec{z})$ can be computed by a $\sum\prod\sum$ circuit Ψ of size at most $d^{O(\sqrt{d})}$[10]. Similar results follow from the application of Theorem 8.

Circuit for f of small depth

Composing the $\sum\prod\sum$ circuit Ψ for $A_d(\vec{z})$ with the circuits of $g_1, g_2, \ldots, g_d \in \mathcal{G}$, we get a circuit Ψ' with the following properties.

- The size of Ψ' is at most $(srn)^{10} \cdot d^{O(\sqrt{d})}$.
- The depth of Ψ' is at most $\Delta + 3$. This follows by combining the bottom $\sum$ layer of the $\sum\prod\sum$ circuit for $A_d(\vec{z})$ with the top $\sum$ layer of the circuits for $g_i \in \mathcal{G}$.
- The degree of Ψ' is at most d^2. This is true because the degree of $A_d(\vec{z})$ is at most d (as argued earlier in this proof), and the degree of every polynomial in $\mathcal{G}$ is at most d (first item in Lemma 17).
- $f = \mathcal{H}_{\leq d}[\Psi'(\vec{x})]$.

To obtain a circuit for f, we apply Lemma 12 to Ψ'. This increases the size of Ψ' by a multiplicative factor of at most $O(d^2)$, while the depth remains the same. This completes the proof of the theorem. ◀

4 Deterministic Identity Testing using Hard Polynomials

In this section, we use Theorem 5 to show that given a family of polynomials which are hard for depth Δ circuits, we can do deterministic identity testing for $\Delta - 5$ circuits in subexponential time. Since the content of this part are very similar to the proofs of similar statements in [10] and [5], we only outline the differences in the proofs (if any), and refer the reader to [5] for details. We start with the following lemma, which is the analog of Lemma 4.1 in [5].

▶ **Lemma 20** (Analog of Lemma 4.1 in [5]). *Let $q(\vec{x}) \in \mathbb{F}[\vec{x}]$ be a (non-zero) polynomial of degree D in n variables, which can be computed by a circuit of size s and depth Δ. Let $m > \log n$ be an integer and let $S_1, S_2, \ldots, S_n \subseteq [\ell]$ be given by Theorem 14, so that $\ell = O(m^2/\log n)$, $|S_i| = m$, and $|S_i \cap S_j| \leq \log n$. For a multilinear polynomial $f \in \mathbb{F}[z_1, z_2, \ldots, z_m]$ of degree d, put*

$$Q(\vec{y}) = Q(y_1, y_2, \ldots, y_\ell) := q\left(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_n})\right) .$$

If $Q(\vec{y}) \equiv 0$, then $f(\vec{z})$ can be computed by an arithmetic circuit of size $O((snD)^{12} d^{O(\sqrt{d})})$ and depth at most $\Delta + 5$.

Note that the bound on the size of f remains non-trivial as long as $d << m$, while the individual degree of q is allowed to be unbounded, whereas the bound in [5] becomes trivial once $\deg_y(q)$ is larger than m.

[10] Instead of Theorem 9, one could use Theorem 8 to get a better size bound than $d^{O(\sqrt{d})}$ at the cost of increasing its depth appropriately. Also, see Remark 1.2. Also, this is one place where the underlying field plays a role, since Theorem 9 is not known to be true over general fields.

Proof Sketch. The proof is along the lines of the proof of Lemma 4.1 in [5]. We now give a sketch of the details. We first define the hybrid polynomials $Q_0(\vec{x},\vec{y}), Q_1(\vec{x},\vec{y}), \ldots, Q_n(\vec{x},\vec{y})$ as follows.

$$Q_j(\vec{x},\vec{y}) = q\left(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_j}), x_{j+1}, x_{j+2}, \ldots, x_n\right) .$$

We know that $Q_0(\vec{x},\vec{y})$ is non-zero, whereas $Q_n(\vec{x},\vec{y})$ is identically zero. Thus, there is an $i \in \{0,1,\ldots,n\}$ such that $Q_i(\vec{x},\vec{y}) \not\equiv 0$ and $Q_{i+1}(\vec{x},\vec{y}) \equiv 0$. We now fix the variables $x_{i+2}, x_{i+3}, \ldots, x_n$ and the variables $\{y_j : j \notin S_{i+1}\}$ to field constants while maintaining the non-zeroness of Q_i. This can be done via Lemma 15. Thus, we have a polynomial $\tilde{q}$ by fixing the aforementioned variables such that the following two conditions hold.

$$\tilde{q}\left(f(\vec{y}|_{S_1\cap S_{i+1}}), f(\vec{y}|_{S_2\cap S_{i+1}}), \ldots, f(\vec{y}|_{S_i\cap S_{i+1}}), x_{i+1}\right) \not\equiv 0 .$$

$$\tilde{q}\left(f(\vec{y}|_{S_1\cap S_{i+1}}), f(\vec{y}|_{S_2\cap S_{i+1}}), \ldots, f(\vec{y}|_{S_i\cap S_{i+1}}), f(\vec{y}|_{S_{i+1}})\right) \equiv 0 .$$

Let $A_0(\vec{y}|_{S_{i+1}}, x_{i+1})$ denote the polynomial

$$\tilde{q}\left(f(\vec{y}|_{S_1\cap S_{i+1}}), f(\vec{y}|_{S_2\cap S_{i+1}}), \ldots, f(\vec{y}|_{S_i\cap S_{i+1}}), x_{i+1}\right) .$$

The above two conditions imply that $f(\vec{y}|_{S_{i+1}})$ is a root of the polynomial $A_0(\vec{y}|_{S_{i+1}}, x_{i+1}) \in \mathbb{F}[\vec{y}|_{S_{i+1}}][x_{i+1}]$, viewed as a polynomial in x_{i+1}. Moreover, $A_0(\vec{y}|_{S_{i+1}}, x_{i+1})$ has a circuit of size at most $O(sn)$ and depth at most $\Delta+2$. This follows from the fact that $f(\vec{y}|_{S_1\cap S_{i+1}})$ is a *multilinear* polynomial in $\log n$ variables, and can thus be computed by a $\sum\prod$ circuit of size at most n. We simply replace the variables $x_1, x_2, \ldots, x_i$ in the circuit for q by these $\sum\prod$ circuits to obtain a circuit for A_0. The degree of A_0 is at most $D\log n$. Finally, Theorem 5 implies that $f(\vec{y}|_{S_{i+1}})$ can be computed by a circuit of size at most $O(\mathsf{poly}(s,n,D)d^{O(\sqrt{d})})$ and depth at most $\Delta+5$, thus completing the proof. ◀

We now sketch the proof of Theorem 3.

Proof Sketch. Once again, the proof follows the proof of Theorems 1 and 2 in [5]. Let $\{f_m\}$ be a family of explicit multilinear polynomials such that f_m has m variables, degree $d \leq O\left(\left(\frac{\log m}{\log\log m}\right)^2\right)$, such that f_m cannot be computed by a circuit of depth Δ and size $\mathsf{poly}(m)$. Let $\varepsilon \in (0, 0.49)$ be an arbitrary constant, and set $m := n^{\varepsilon}$, and $f = f_m$.

Given as input a circuit $C \in \mathbb{F}[\vec{x}]$ of size s, depth $\Delta - 5$ and degree D on n variables, let $q \in \mathbb{F}[\vec{x}]$ be the polynomial computed by C. The goal here is to determine whether q is nonzero. From the equivalence of black-box PIT and hitting set, it suffices to construct hitting set for circuit class of the above properties.

- We construct a design $S_1, S_2, \ldots, S_n \subseteq [\ell]$ using Theorem 14 where each set S_i has size m, $\ell = O(m^2/\log n) \leq n^{2\varepsilon} < n^{0.98}$ and $|S_i \cap S_j| \leq \log n$. This can be done in deterministic time $2^{O(n^{2\varepsilon})}$.
- We pick a subset T of the field $\mathbb{F}$ of size $Dd+1$ and evaluate the polynomial $q\left(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_n})\right)$ on all points of T^{ℓ}. $H = \{(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_n})) \mid \vec{y} \in T^{\ell}\}$ is then our candidate hitting set of size $(Dd+1)^{\ell} = n^{O(n^{2\varepsilon})} < n^{O(n^{0.98})}$. Note that the set can be constructed deterministically in time $m^d \cdot n^{O(n^{2\varepsilon})} = n^{O(n^{2\varepsilon})}$.

We now argue about the correctness, *i.e.,* q does not vanish on the hitting set if and only if q is not identically zero. Observe that if the polynomial $q\left(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_n})\right)$ is not identically zero, then it has degree at most Dd and hence by Lemma 15, q does not vanish on the set H. Else, $q\left(f(\vec{y}|_{S_1}), f(\vec{y}|_{S_2}), \ldots, f(\vec{y}|_{S_n})\right) \equiv 0$. But then, by Lemma 20, we get that f can be computed by a circuit of depth Δ and size at most $O\left(\mathsf{poly}(s,n,D)d^{O(\sqrt{d})}\right)$.

If s, D are $\mathsf{poly}(n)$, then this bound is $\mathsf{poly}(m)$ which contradicts the assumed hardness of $f = f_m$ for circuits of depth Δ. This shows that H is a hitting set for the desired circuit class and completes the proof. ◀

References

1 Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008. doi:10.1109/FOCS.2008.32.

2 Walter Baur and Volker Strassen. The complexity of partial derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. doi:10.1016/0304-3975(83)90110-X.

3 Peter Bürgisser. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, 2004. doi:10.1007/s10208-002-0059-5.

4 Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. *CoRR*, abs/1710.03214, 2017. URL: http://arxiv.org/abs/1710.03214.

5 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. doi:10.1137/080735850.

6 Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 451–465. IEEE Computer Society, 2015. doi:10.1109/FOCS.2015.35.

7 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 128–135. ACM, 2014. doi:10.1145/2591796.2591824.

8 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 578–587. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.68.

9 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. doi:10.1145/2629541.

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

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

12 Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.

13 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153. ACM, 2014. doi:10.1145/2591796.2591847.

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

15 Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 19:1–19:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.19.

16 Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 31:1–31:30. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.31.

17 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 364–373. IEEE Computer Society, 2014. doi:10.1109/FOCS.2014.46.

18 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. doi:10.1016/S0022-0000(05)80043-1.

19 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997. doi:10.1007/BF01294256.

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

21 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.

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

23 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.

24 Ran Raz, Amir Shpilka, and Amir Yehudayoff. A lower bound for the size of syntactically multilinear arithmetic circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. doi:10.1137/070707932.

25 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.

26 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, https://github.com/dasarpmar/lowerbounds-survey/, 2016. URL: https://github.com/dasarpmar/lowerbounds-survey/releases/.

27 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

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