THEORY OF COMPUTING, Volume 15 (13), 2019, pp. 1–34
www.theoryofcomputing.org

Closure Results for Polynomial Factorization

Chi-Ning Chou* Mrinal Kumar Noam Solomon

Received July 7, 2018; Revised January 30, 2019; Published November 1, 2019

Abstract: In a sequence of fundamental results in the 1980s, Kaltofen (SICOMP 1985, STOC'86, STOC'87, RANDOM'89) showed that factors of multivariate polynomials with small arithmetic circuits have small arithmetic circuits. In other words, the complexity class VP is closed under taking factors. A natural question in this context is to understand if other natural classes of multivariate polynomials, for instance, arithmetic formulas, algebraic branching programs, bounded-depth arithmetic circuits or the class VNP, are closed under taking factors.

In this paper, we show that all factors of degree $\log^a n$ of polynomials with $\mathsf{poly}(n)$-size depth-k circuits have $\mathsf{poly}(n)$-size circuits of depth $O(k+a)$. This partially answers a question of Shpilka–Yehudayoff (Found. Trends in TCS, 2010) and has applications to hardness–randomness tradeoffs for bounded-depth arithmetic circuits.

As direct applications of our techniques, we also obtain simple proofs of the following results.

- The complexity class VNP is closed under taking factors. This confirms Conjecture 2.1 in Bürgisser's monograph (2000) and improves upon a recent result of Dutta, Saxena and Sinhababu (STOC'18) who showed a quasipolynomial upper bound on the number of auxiliary variables and the complexity of the verifier circuit of factors of polynomials in VNP.

A preliminary version of this paper, titled "Hardness vs Randomness for Bounded Depth Arithmetic Circuits," appeared in the Proceedings of the 33rd Computational Complexity Conference, 2018.

*Supported by Boaz Barak's NSF awards CCF 1565264 and CNS 1618026.

ACM Classification: F.1.3

AMS Classification: 68Q15, 68Q17

Key words and phrases: algebraic complexity, polynomial factorization circuit lower bounds, Polynomial Identity Testing, Polynomial Identity Lemma

DOI: 10.4086/toc.2019.v015a013

- A factor of degree d of a polynomial P which can be computed by an arithmetic formula (or an algebraic branching program) of size s has a formula (an algebraic branching program, resp.) of size $\mathsf{poly}(s, d^{\log d}, \deg(P))$. This result was first shown by Dutta et al. (STOC'18) and we obtain a slightly different proof as an easy consequence of our techniques.

Our proofs rely on a combination of the ideas, based on *Hensel lifting*, developed in the polynomial factoring literature, and the depth-reduction results for arithmetic circuits, and hold over fields of characteristic zero or of sufficiently large characteristic.

1 Introduction

A fundamental question in computational algebra is the question of polynomial factorization: Given a polynomial P, can we efficiently compute the factors of P? In this paper, we will be interested in the following closely related question: Given a *structured* polynomial P, what can we say about the structure of factors of P?

In a sequence of seminal papers, Kaltofen [15, 16, 17, 18] showed that if a polynomial P of degree d in n variables has an arithmetic circuit of size s, then each of its factors has an arithmetic circuit of size $\mathsf{poly}(s,n,d)$. Moreover, he also showed that given the circuit for P, the circuits for its factors can be computed in time $\mathsf{poly}(s,n,d)$ by a randomized algorithm.

Another way of stating this result is that the complexity class VP, which we now define, is *uniformly closed under taking factors*.

Definition 1.1 (VP)**.** A family $\{f_n\}$ of polynomials over a field $\mathbb{F}$ is said to be in the class $\mathsf{VP}_{\mathbb{F}}$ if there exist polynomially bounded functions $d,k,v : \mathbb{N} \to \mathbb{N}$ and a circuit family $\{g_n\}$ such that $\deg(f_n) \leq d(n)$, $\text{size}(g_n) \leq s(n)$, and f_n is computed by g_n for every sufficiently large $n \in \mathbb{N}$.

We remark that factorization is a fundamental algebraic notion, and so closure under factorization indicates that a complexity class is algebraically nice in some sense. Thus, it is a natural question to ask if any of the other naturally and frequently occurring classes of polynomials like VF (polynomials with small formulas), VBP (polynomials with small algebraic branching programs), bounded-depth arithmetic circuits, or the class VNP (the algebraic analog of NP or #P) are closed under taking factors.

In recent years, we have had some progress on the question of closure under factorization for bounded-depth arithmetic circuits (see [8, 29]) or the classes VF, VBP and VNP (see [7]). We will discuss these results in a later part of this section.

In addition to being basic questions in algebraic complexity, some of these closure results also have applications to extending the hardness vs. randomness framework of Kabanets and Impagliazzo [13] to formulas, branching programs or bounded-depth arithmetic circuits. Indeed, Kaltofen's closure result for arithmetic circuits is crucial ingredient in the proof of Kabanets and Impagliazzo [13].

1.1 Hardness and randomness

Two of the most basic questions in algebraic complexity theory are the question of proving superpolynomial lower bounds on the size of arithmetic circuits computing some explicit family of polynomials[1] and that of designing efficient deterministic algorithms for Polynomial Identity Testing (PIT).

The progress on these questions for general arithmetic circuits has been painfully slow. To date, there are no non-trivial algorithms for PIT for general arithmetic circuits, while the best known lower bound on the circuit size for explicit families of polynomials, due to Bauer and Strassen [3], is a slightly superlinear lower bound $\Omega(n \log n)$, proved 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 circuit lower bounds known are just slightly superlinear [32].

In a very influential work, Kabanets and Impagliazzo [13] 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 [27] to the algebraic setting. In their proof, Kabanets and Impagliazzo combine the use of the Nisan generator [26] with Kaltofen's result that all factors of a low-degree (degree $\mathsf{poly}(n)$) polynomial with a $\mathsf{poly}(n)$-size circuit are computable by $\mathsf{poly}(n)$-size circuits [18]. They showed that given an explicit family of *hard* polynomials, one can obtain a *non-trivial*[2] deterministic algorithm for PIT.

The extremely slow progress on the circuit lower bound and PIT questions for general circuits has led to a lot of attention on understanding these questions for more structured subclasses of arithmetic circuits. Arithmetic formula [14], algebraic branching programs [21], multilinear circuits [31, 35, 34], and bounded-depth arithmetic circuits [28, 32, 11, 10, 23] are some examples of such circuit classes. An intriguing question is to ask if the equivalence of PIT and lower bounds also carries over to these more structured circuit classes. For example, do superpolynomial 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 [13], and extending the approach of Kabanets and Impagliazzo to answer these questions seems to be intimately related to the questions about closure of arithmetic formulas and bounded-depth circuits under polynomial factorization.

We now describe our results, and discuss how they relate to prior work.

2 Results and prior work

2.1 Factors of polynomials with bounded-depth circuits

For our first set of results, we study the bounded-depth circuit complexity of factors of polynomials which have small bounded-depth circuits. We prove the following result.

Theorem 2.1. *Let $\mathbb{F}$ be a field of characteristic zero. Let $P \in \mathbb{F}[\mathbf{x}]$ be a polynomial of degree r in n variables that can be computed by an arithmetic circuit of size s and depth Δ. Let $f \in \mathbb{F}[\mathbf{x}]$ be an*

[1] Informally, a family $\{f_n\}$ of polynomials is said to be explicit, if there is a deterministic algorithm which takes an input an $n \in \mathbb{N}$ and a monomial and outputs the coefficient of the monomial in f_n. Moreover, the time complexity of the algorithm is polynomially bounded in n and the degree of the monomial. See Section 4.5 for a formal definition.

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

irreducible polynomial of degree d such that f divides P. Then f can be computed by a circuit of depth $\Delta + O(1)$ and size $\mathsf{poly}(s,r,n) \cdot d^{O(\sqrt{d})}$. Furthermore, for any $k \in \mathbb{N}$, f can be computed by a circuit of depth $\Delta + O(k)$ and size $\mathsf{poly}(s,r,n) \cdot d^{O(d^{1/k})}$.

Thus, low-degree factors of polynomials with small shallow circuits have small shallow circuits. This partially answers an open problem (Open Problem 19 in [38]) of Shpilka and Yehudayoff who asked whether the factors of a multivariate polynomial with a small shallow circuit can also be computed by small shallow circuits.

Our proof gives a smooth tradeoff between the depth of the circuit for the factor and its size. The tradeoff is governed by the depth-reduction results for arithmetic circuits (see Theorem 4.3). We remark that the result is also true when the characteristic of the underlying field is sufficiently large. The result in the literature, which is most closely related to Theorem 2.1, is due to Oliveira [29]. He studied the question of bounded-depth circuit complexity of factors of polynomials with small bounded-depth circuits, for polynomials of low individual degree. He showed that if a polynomial P of individual degree r is computable by a circuit of size s and depth Δ, then every factor of P of degree d can be computed by a circuit of size $\mathsf{poly}(s,r,d^r)$ and depth $\Delta + 5$. Thus, for polynomials with small individual degree, the results in [29] are strictly better than ours, whereas for polynomials with unbounded individual degree, we get a better upper bound on the complexity of factors of total degree $\mathsf{poly}(\log n)$.

One of our main motivations for studying this question is the connection to hardness-randomness tradeoffs for bounded-depth arithmetic circuits. In the next section, we describe the implications of our results in this context.

2.2 Hardness vs. randomness for bounded-depth circuits

Dvir, Shpilka and Yehudayoff [8] initiated the study of the question of the equivalence between PIT and lower bounds for bounded-depth circuits. Dvir et al. observed that a part of the proof in [13] 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 superpolynomial lower bounds for depth-Δ arithmetic circuits imply 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 interest: Does a superpolynomial lower bound for depth-Δ arithmetic circuits imply non-trivial deterministic PIT for depth-Δ' arithmetic circuits?[3] In particular, can we get rid of the "bounded individual degree" condition from the results in [8]?

In this paper, we partially answer this question in the affirmative. Here is an informal statement of the result.

Theorem 2.2 (Informal)**.** *A superpolynomial 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 of degree at most $O(\log^2 n / \log^2 \log n)$. Thus, by strengthening the hardness hypothesis in [8], we remove the bounded individual degree restriction from the implication. We now state the result in Theorem 2.2 formally.

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

Theorem 2.3. *Let $\Delta \geq 6$ be a positive integer, and let $\varepsilon > 0$ be any real number. Let $\{f_m\}$ be an explicit family of 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 $C \in \mathbb{Q}[\mathbf{x}]$ 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.*

Some remarks on the above theorem statement.

Remark 2.4. The running time of the PIT algorithm gets better as the lower bound gets stronger. Also, the constraint on the degree of family of hard polynomials can be further relaxed a bit, at the cost of strengthening the hardness assumption, and increasing the running time of the resulting PIT algorithm.[4] We leave it to the interested reader to work out these details. We also note that the multilinearity assumption on the family of hard polynomials is without loss of generality.

As discussed earlier, Theorem 2.3 is closely related to the main result in [8]. We now discuss their similarities and differences.

Comparison with Dvir et al. [8].

- **Degree constraint on the hard polynomial.** While Theorem 2.3 requires that the hard polynomial on m variables has degree $O(\log^2 m / \log^2 \log m)$, Dvir et al. [8] did not have a similar constraint.

- **Individual degree constraint for PIT.** In [8], the authors get PIT for shallow circuits with bounded individual degree, whereas our Theorem 2.3 does not make any assumptions on individual degrees in this context.

The key technical challenge for extending the known hardness-randomness tradeoffs for general circuits [13] to restricted circuit classes like formulas or bounded-depth circuits is the following question: Let $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial of degree r and let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial of degree d such that $P(\mathbf{x}, f) \equiv 0$. Assuming P can be computed by a shallow circuit (or arithmetic formula) of size s, can f be computed by a shallow circuit (or arithmetic formula) of size $\mathsf{poly}(s, n, d, r)$?

In [8], the authors partially answer this question by showing that the polynomial f can be computed by a shallow circuit of size $\mathsf{poly}(s, r, d^{\deg_y(P)})$. Thus, for the case of polynomials P which have small individual degree with respect to y, they answer the question in the affirmative.

Our main technical observation, which we state next, gives an upper bound on the shallow circuit complexity of polynomials $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ of *low degree* such that there is a polynomial $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ with small shallow circuits satisfying $P(\mathbf{x}, f) = 0$. In other words, if we view P as a univariate polynomial in y with coefficients coming from the ring $\mathbb{F}[\mathbf{x}]$, then f is a low-degree polynomial that is a root of P. We now state the theorem.

[4] If we assume subexponential lower bound, then we can get a quasipolynomial time PIT. Note that this is the parameter region used in [8].

Theorem 2.5. *Let $\mathbb{F}$ be a field of characteristic zero. Let $P \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial of degree r in $n+1$ variables that can be computed by an arithmetic circuit of size s and depth Δ. Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial of degree d such that*

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

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

Furthermore, for any natural number k, f can be computed by a circuit of depth $\Delta + O(k)$ and size $O((srn)^{10} d^{O(d^{1/k})})$.

We conclude this section with a short discussion on the *low-degree* condition in the hypothesis of Theorem 2.3.

2.2.1 The low-degree condition

The *low-degree* condition in the hypothesis of Theorem 2.3 appears to be extremely restrictive. It is natural to wonder if the question of proving superpolynomial lower bounds for bounded-depth circuits for an explicit family of polynomials of *low degree* is much harder than the question of proving superpolynomial lower bounds for bounded-depth circuits for an explicit family of polynomials of potentially larger degree?[5] Currently, we do not even know quadratic lower bounds for arithmetic circuits of bounded 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 shallow circuits carry over to the low-degree regime. For instance, the proofs of superpolynomial lower bounds for homogeneous depth-3 circuits by Nisan and Wigderson [28], superpolynomial lower bounds for homogeneous depth-4 circuits based on the idea of shifted partial derivatives (see for example, [11, 19, 10, 23]) and superlinear lower bound due to Raz [32] 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 [22] 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 regimes is a separation between homogeneous depth-5 and homogeneous depth-4 circuit [22] 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 [33], 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 superpolynomial lower bound for arithmetic formulas, provided $d \le O(\log n / \log\log n)$. As far as we know, we do not know of such connections in the high-degree regime.

One prominent family of lower bound results which do not seem to generalize to this low-degree regime are the superpolynomial lower bounds for multilinear formulas [31], and multilinear bounded-depth circuits [35]. In fact, the results in [33] show that superpolynomial lower bounds for set multilinear formulas[6] for polynomials of degree $O(\log n / \log\log n)$ imply superpolynomial lower bounds for general arithmetic formulas.

[5] In general, the degree only has to be bounded by a polynomial function in the number of variables.

[6] Set multilinear formulas are a more structured sub-class of multilinear formulas and are a very natural model of computation in certain settings. See [33] for a formal definition.

In the context of polynomial factorization, low-degree factors of polynomials with small circuits have been considered before. For instance, Forbes [9] gave a quasipolynomial-time deterministic algorithm to test if a given polynomial of bounded 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.

2.3 Factors of polynomials in VNP

We start by formally defining the complexity class VNP.

Definition 2.6 (VNP)**.** A family of polynomials $\{f_n\}$ over a field $\mathbb{F}$ is said to be in the class $\mathsf{VNP}_{\mathbb{F}}$ if there exist polynomially bounded functions $k, w, v : \mathbb{N} \to \mathbb{N}$ and a family $\{g_n\}$ in $\mathsf{VP}_{\mathbb{F}}$ such that for every sufficiently large $n \in \mathbb{N}$,

$$f_n(x_1, x_2, \ldots, x_{k(n)}) = \sum_{\mathbf{y} \in \{0,1\}^{w(n)}} g_{v(n)}\left(x_1, x_2, \ldots, x_{k(n)}, y_1, y_2, \ldots, y_{w(n)}\right).$$

We refer to the y variables in the definition above as auxiliary variables, and the polynomial family g_n as the family of verifier polynomials. Essentially, VNP can be thought of as the algebraic analog of NP, and understanding if VNP is different from VP is the algebraic analog of the famous P vs. NP question. As discussed earlier in this section, Kaltofen's closure result for VP does not seem to immediately extend to VNP, and whether or not the factors of polynomials in VNP are in VNP was an open question. Bürgisser made the following conjecture [4, Conjecture 2.1].

Conjecture 2.7 (Bürgisser)**.** *The class* VNP *is closed under taking factors.*

As a direct application of our proof of Theorem 2.1, we confirm this conjecture over fields of characteristic zero or of sufficiently large characteristic. We obtain a simple proof of the following statement.

Theorem 2.8 (Informal)**.** *The class* VNP *is closed under taking factors.*

The main technical statement which immediately gives us this closure result is the following theorem.

Theorem 2.9. *Let $\mathbb{F}$ be a field of characteristic zero. Let $P(\mathbf{x})$ be a polynomial of degree r over $\mathbb{F}$, and let $Q(\mathbf{x}, \mathbf{y})$ be a polynomial in $n+m$ variables such that*

$$P(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^m} Q(\mathbf{x}, \mathbf{y}),$$

and Q can be computed by a circuit of size s. Let f be any irreducible factor of P of degree d. Then, there exists an $m' \leq \mathsf{poly}(s, r, d, n, m)$ and polynomial $h(x_1, x_2, \ldots, x_n, z_1, z_2, \ldots, z_{m'})$ where $h(\mathbf{x}, \mathbf{z})$ can be computed by a circuit of size $s' \leq \mathsf{poly}(s, r, d, n, m)$ such that

$$f(\mathbf{x}) = \sum_{\mathbf{z} \in \{0,1\}^{m'}} h(\mathbf{x}, \mathbf{z}).$$

We remark that in the proof of the above theorem, our techniques can be replaced by analogous statements from [8, 29]. Although this is a simple observation, this does not appear to have been noticed prior to this work. The best upper bound on the complexity of factors of polynomials in VNP in prior work is a recent result of Dutta, Saxena, Sinhababu [7], who showed a bound of $\mathsf{poly}(n,r,s,m,d^{O(\log d)})$ on the number of auxiliary variables and the circuit complexity of verifier polynomials h.

As an easy consequence of our proofs, we also obtain another (slightly different) proof of the following result of Dutta et al. [7].

Theorem 2.10 (Dutta, Saxena, Sinhababu)**.** *Let $P(\mathbf{x})$ be a polynomial of degree r in n variables which can be computed by an arithmetic formula (or algebraic branching program) of size s, and let $f(\mathbf{x})$ be a factor of P of degree d. Then, $f(\mathbf{x})$ can be computed by an arithmetic formula (algebraic branching program, resp.) of size* $\mathsf{poly}(s,r,n,d^{O(\log d)})$.

3 Proof overview

The key technical ingredients of our results in this paper is Theorem 2.5. We start by describing the main steps in its proof.

Proof sketch of Theorem 2.5. Our proof of Theorem 2.5 follows the outline of the proof of the analogous theorem about the structure of roots in [8]. 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 $P(\mathbf{x},y)$. More formally, in the k^{th} step, we start with a polynomial h_k which agrees with f on all monomials of degree k, and use it to obtain a polynomial h_{k+1} which agrees with f on all monomials of degree $k+1$. 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 d. This lifting step is exactly the same as that in [8] or in some of the earlier works on polynomial factorization [5], and is formally stated in Lemma 5.1.

The key insight of Dvir et al. [8] was that if $\deg_y(P) = t$, and $C_0(\mathbf{x}), C_1(\mathbf{x}), \ldots, C_t(\mathbf{x})$ are polynomials such that $P(\mathbf{x},y) = \sum_{i=1}^{t} C_i(\mathbf{x})y^t$, then for every $k \in \{0,1,\ldots,d\}$, we have a polynomial B_k of degree k such that

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

Now, consider the case when $t \ll n$ (for instance $t = O(1)$). It follows from standard interpolation results for shallow circuits (see Lemma 4.9) that each of the polynomials $C_i(\mathbf{x})$ has a circuit of size $O(sr)$ and depth Δ since P has a polynomial of size s and depth Δ. Thus, $h_d(\mathbf{x})$ can be written as a sum of $\binom{d+t}{t} = O(d^t)$ 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 4.9), 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 $g_0(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_d(\mathbf{x})$ of degree d, and polynomials $A_0, A_1, \ldots, A_k$ on $d+1$ variables such that for every $k \in \{0, 1, \ldots, d\}$,

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

Moreover, for every k, A_k has degree k and is computable by a circuit of size $O(d^3)$. 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 subexponential 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 $d^{O(\sqrt{d})}$, using the depth-reduction result of [12].[7]

One point to note is that just from Kaltofen's result [18], it follows that f has an arithmetic circuit[8] of size $\mathsf{poly}(n)$. Thus, from Theorem 4.4, it follows that f has a circuit of depth-3 of size $n^{O(\sqrt{d})}$. The key advantage of Theorem 2.5 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 $\mathsf{poly}(n)$.

Proof sketch of Theorem 2.1. To get Theorem 2.1 from Theorem 2.5, we also have to upper bound the complexity of factors which are not of the form $y - f(\mathbf{x})$, i. e., are non-linear in every variable. This involves the use of some standard techniques in this area. We first preprocess P such that it is monic in y, and then we work over the algebraic closure of the field $\mathbb{F}[\mathbf{x}]$, and view P as a univariate in y over this field. We then use Lemma 5.1 to approximate these roots by polynomials, and eventually combine them using Lemma 6.3 from [29] to obtain the factor f. We get bounds on the circuit size and depth of the factor f by keeping tab on the growth of these parameters in each step of the outlined algorithm.

Proof sketch of Theorem 2.3. Theorem 2.5, when combined with the standard machinery of Nisan-Wigderson designs immediately yields Theorem 2.3.

Proof sketch of Theorem 2.9. For the proof of Theorem 2.9, we follow the same outline as above to conclude that every factor f of a polynomial $P = \sum_{\mathbf{y} \in \{0,1\}^m} Q(\mathbf{x}, \mathbf{y})$ can be written as

$$f(\mathbf{x}) = \mathcal{H}_{\leq d}\left[B(g_0(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_d(\mathbf{x}))\right],$$

where B has a circuit of size $\mathsf{poly}(d)$ and degree d and each polynomial g_i can be expressed as $\sum_{\mathbf{y} \in \{0,1\}^{m'}} \tilde{Q}_i(\mathbf{x}, \mathbf{y})$, where the number of auxiliary variables m' and the circuit size of Q are each less than $\mathsf{poly}(s, n, m, d, r)$, where s is the circuit size of Q, r is the degree of P. The proof follows from a result of Valiant [40], where he showed that compositions such as $B(g_0(\mathbf{x}), g_1(\mathbf{x}), \ldots, g_d(\mathbf{x}))$ can be written in the form $\sum_{\mathbf{y} \in \{0,1\}^{m'}} Q'(\mathbf{x}, \mathbf{y})$ with m'' and the circuit complexity of Q' being $\mathsf{poly}(s, n, m, d, r)$.

[7] See Theorem 4.4 for a formal statement of this result.
[8] Of potentially very large depth.

Note that composing B and g_i into the above form is not straightforward since direct replacement of g_0 with $\tilde{Q}_i$ might not work.[9] For completeness, we include a proof of this using the depth-reduction results in [41]. (See Theorem 8.2 and Claim 8.4 and the appendix for the proof.).

We remark that the proof outlined above bounds the complexity of the factor f once at the end of the lifting, whereas in [7], the authors prove an upper bound on the number of auxiliary variables and the circuit complexity of the verifier circuit for the approximation of the factor of P at the end of each step of the lifting process. They show that in every step of lifting, these parameters grow only by a multiplicative factor of d^2, and there are $O(\log d)$ steps of lifting in total, hence the total blowup of $d^{O(\log d)}$ in the process. In contrast, we get a polynomial upper bound on the blowup in the number of auxiliary variables, and the circuit size of the verifier circuit for the factor f, by a one step analysis.

Another crucial point to note is that Theorem 2.9 also follows if in the approach outlined above, we replace our structure theorem for the structure of low-degree factors by an analogous statement in [8] and [29]. This is because, the degree of the factor we are seeking and the depth of the circuit obtained for the factor do not play a critical role in this proof as long as they are not too large. Thus, closure of VNP under taking factors follows from the results known prior to this work, although as far as we know, this does not seem to have been noticed before.

4 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.

4.1 Notation

- We use boldface letters $\mathbf{x},\mathbf{y},\mathbf{z}$ to denote a list of variables.

- For a function $s:\mathbb{N}\to\mathbb{N}$, we say that $s(n)\leq \mathsf{poly}(n)$, if there are constants $n_0,a\in\mathbb{N}$ such that $\forall n>n_0$, $s(n)\leq n^a$.

- For a (multivariate) polynomial P, $\deg(P)$ denotes the total degree of P and $\deg_y(P)$ denotes the degree of P with respect to the variable y.

- Let $P\in\mathbb{F}[\mathbf{x}]$ be a polynomial. For every $k\in\mathbb{N}$, $\mathcal{H}_k[P]$ denotes the homogeneous component of P of degree k. Similarly, $\mathcal{H}_{\leq k}[P]$ is defined to be equal $\sum_{i=0}^{k}\mathcal{H}_i[P]$.

- We say that a polynomial f is a factor of a polynomial P of multiplicity equal to m, if f^m divides P, and f^{m+1} does not divide P.

[9] Consider the following toy example: Let B be a multiplication gate with two inputs from the same subcircuit $g_0(\mathbf{x})$, i. e., $B(g_0(\mathbf{x}))=g_0(\mathbf{x})^2$. However, if we directly replace $g_0(\mathbf{x})$ with $\tilde{Q}_0$, we would get $\sum_{y\in\{0,1\}^{m'}}\tilde{Q}_i(\mathbf{x},\mathbf{y})^2$, which might not be $g_0(\mathbf{x})$.

4.2 Arithmetic circuits

Definition 4.1 (Arithmetic circuits)**.** An arithmetic circuit Ψ over a field $\mathbb{F}$ and variables $\mathbf{x} = (x_1, \ldots, x_n)$ is a directed acyclic graph, the vertices of which we refer to as gates. The gates of in-degree zero (or input gates) are labeled by elements in $\mathbb{F}$ and variables in $\mathbf{x}$, and the internal gates are labeled by $+$ (sum gates) or $\times$ (product gates). The gates of out-degree zero in Ψ are called output gates. The circuit Ψ computes a polynomial in $\mathbb{F}[\mathbf{x}]$ in a natural way: the input gates compute the polynomial equal to its label. A sum gate computes the polynomial equal to the sum of the polynomials computed at its inputs, while a product gate computes the polynomial equal to the product of the polynomials computed at its inputs.

For an arithmetic circuit Ψ, we use size(Ψ) to denote the number of edges in Ψ. The depth of Ψ is the length of the longest path from any input gate to any output gate. Throughout this paper, we assume that all our circuits are layered with alternating layers of addition and multiplication gates, with the input gates forming the bottom layer and the output gates forming the top layer. The directed edges should be thought of as pointing upward. 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 $\sum\prod\sum\prod$. Let P be the polynomial computed by Ψ. For every $k \in \mathbb{N}$, we use $\mathcal{H}_k[\Psi]$ to denote the homogeneous component of P of degree k. Similarly, $\mathcal{H}_{\leq k}[\Psi]$ is defined to be equal $\sum_{i=0}^{k} \mathcal{H}_i[\Psi]$.

4.3 Taylor's expansion

A crucial tool for our proofs is the following classical lemma.

Lemma 4.2 (Taylor's expansion)**.** *Let $P(y) \in \mathbb{F}[y]$ be a polynomial of degree d. Then,*

$$P(y+z) = P(y) + z \cdot P^{(1)}(y) + z^2 \cdot \frac{P^{(2)}(y)}{2!} + \cdots + z^d \cdot \frac{P^{(d)}(y)}{d!},$$

where, for every k, $P^{()}(y) = \frac{\partial^k P(y)}{\partial y^k}$ is the derivative of order k of P with respect to y.

At a later point in the paper, we work with multivariate polynomials $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ and view them as univariate polynomials in y with the coefficients coming from the field of fractions $\mathbb{F}(\mathbf{x})$. In this case, the derivatives $P^{(k)}(y) = \frac{\partial^k P}{\partial y^k}$ as defined above are elements of $\mathbb{F}[\mathbf{x}]$.

4.4 Depth reduction

We will use the following depth-reduction theorems as a black boxes for our proofs. The first result is by Agrawal–Vinay [1], Koiran [20], and Tavenas [39].

Theorem 4.3 (Depth reduction to depth $(2k)$)**.** *Let k be a positive integer and $\mathbb{F}$ be any field. If $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ is an n-variate polynomial of degree d that be computed by an arithmetic circuit Ψ of size s, then P can be computed by a depth-$(2k)$ circuit of size $(snd)^{O(d^{1/k})}$.*

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, due to Gupta, Kamath, Kayal, and Saptharishi [12], gives a further reduction to depth 3, as long as the field is of characteristic zero, and will be useful for our proof.

Theorem 4.4 (Depth reduction to depth 3)**.** *Let $\mathbb{F}$ be a field of characteristic zero. Let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be an n-variate polynomial of degree d that can be computed by an arithmetic circuit Ψ of size s. Then, P can be computed by a $\sum\prod\sum$ circuit of size $(snd)^{O(\sqrt{d})}$.*

We will also need the following two results which give formula upper bounds for polynomials with small circuits. The results immediately follow from a classical depth-reduction result of Valiant, Skyum, Berkowitz, and Rackoff [41].

Theorem 4.5 (Valiant et al.)**.** *Let $P(\mathbf{x})$ be a polynomial of degree d in n variables which can be computed by a circuit C of size s. Then, P can also be computed by a homogeneous circuit C' of size $\mathsf{poly}(s,n,d)$, with the following properties.*

- *Every product gate in C' has fan-in at most 5.*
- *For every product gate g in C', the degree of the polynomial computed by any child of g is at most half of the degree of the polynomial computed at g.*
- *C' has alternating layers of sum and product gates, where the sum fan-ins can be unbounded.*

Theorem 4.6 (Valiant et al.)**.** *Let $P(\mathbf{x})$ be a polynomial of degree d in n variables which can be computed by a circuit of size s. Then, P can also be computed by a formula of size $(sn)^{O(\log d)}$.*

4.5 Explicit family of polynomials

The following definition is from Dvir, Shpilka, and Yehudayoff [8].

Definition 4.7 (Dvit et al.)**.** Let $\{f_m\}$ be a family of multilinear polynomials such that $f_m \in \mathbb{Q}[x_1,\ldots,x_m]$ 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 coefficients of f_m in time $2^{O(m)}$.

4.6 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 4.8 (Homogenization)**.** *Let $\mathbb{F}$ be any field, and let $\Psi \in \mathbb{F}[\mathbf{x}]$ be an arithmetic circuit of size s. Then, for every $k \in \mathbb{N}$, there is a homogeneous circuit Ψ_k of formal degree k and size $O(k^2 s)$, such that*

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

Theorem 4.8 gives us a way of extracting homogeneous components of the polynomial computed by a given circuit. We also need the following related well known lemma, whose proof we briefly sketch.

Lemma 4.9 (Interpolation)**.** *Let $\mathbb{F}$ be any field with at least $d+1$ elements. Let $P(\mathbf{x},y) \in \mathbb{F}[\mathbf{x},y]$ be a polynomial of degree d. Let $C_0(\mathbf{x}), C_1(\mathbf{x}), \ldots, C_d(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be polynomials such that $P(\mathbf{x},y) = \sum_{j=0}^{d} y^d \cdot C_j(\mathbf{x})$. Then, if $P(\mathbf{x},y)$ has a circuit of size s and depth Δ, then for every $j \in \{0,1,\ldots,d\}$, $C_j(\mathbf{x})$ has a circuit of size $O(sd)$ and depth Δ.*

Proof sketch. For the proof, we view P as a univariate polynomial of degree d in y with coefficients in the ring $\mathbb{F}[\mathbf{x}]$. Thus, each C_j can be written as an appropriate linear combination of $P(\mathbf{x},\alpha_0)$, $P(\mathbf{x},\alpha_1)$, $\ldots$, $P(\mathbf{x},\alpha_d)$, where $\alpha_0, \alpha_1, \ldots, \alpha_d$ are distinct elements of the field $\mathbb{F}$. Observe that for every $\alpha \in \mathbb{F}$, $P(\mathbf{x},\alpha)$ has a circuit of size s and depth Δ. To compute C_j, we have to take an appropriate linear combination of these circuits, but the linear combination can be absorbed in the top sum gate, and hence this process does not incur an increase in depth, while the size grows by a factor of at most d. $\square$

The following corollary of Lemma 4.9 would also be useful for us. The proof follows immediately from the proof of Lemma 4.9.

Lemma 4.10 (Interpolation for formulas)**.** *Let $\mathbb{F}$ be any field with at least $d+1$ elements. Let $P(\mathbf{x},y) \in \mathbb{F}[\mathbf{x},y]$ be a polynomial of degree d. Let $C_0(\mathbf{x}), C_1(\mathbf{x}), \ldots, C_d(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be polynomials such that $P(\mathbf{x},y) = \sum_{j=0}^{d} y^d \cdot C_j(\mathbf{x})$. Then, if $P(\mathbf{x},y)$ has a* formula *of size s, then for every $j \in \{0,1,\ldots,d\}$, $C_j(\mathbf{x})$ has a* formula *of size $O(sd)$.*

4.7 Hitting sets

Definition 4.11. 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 $\mathbf{a} \in \mathcal{P}$ such that $C(\mathbf{a}) \neq 0$.

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 *black-box*, 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 *black-box* in this sense.

4.8 Nisan designs

We state the following well known result of Nisan [26] on the explicit construction of combinatorial designs.

Theorem 4.12 (Nisan)**.** *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 $i \neq j$, $\left|S_i \cap S_j\right| \leq \log n$.*
- *$\ell = O(\frac{m^2}{\log n})$.*

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

4.9 The Polynomial Identity Lemma

We now state the well-known Polynomial Identity Lemma.[10]

Lemma 4.13 (Polynomial Identity Lemma)**.** *Let $\mathbb{F}$ be a field, and let $P \in \mathbb{F}[\mathbf{x}]$ be a non-zero polynomial of degree (at most) d in n variables. Then, for any finite set $S \subset \mathbb{F}$ we have*

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

In particular, if $|S| \geq d+1$, then there exists some $\mathbf{a} \in S^n$ satisfying $P(\mathbf{a}) \neq 0$. 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 d in n variables is identically zero.

5 Low-degree roots of polynomials with shallow circuits

In this section, we prove Theorem 2.5, which is also our main technical observation. 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, which appears in many of the prior works in this area including [8]. The statement and the proof below, are from Dvir et al. [8].

Lemma 5.1 (Hensel Lifting [8])**.** *Let $P \in \mathbb{F}[\mathbf{x}, y]$ and $f \in \mathbb{F}[\mathbf{x}]$ be polynomials such that $P(\mathbf{x}, f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\mathbf{x}, f(\mathbf{x}))\right] = \delta \neq 0$. Let $i \in \{1, 2, \ldots, \deg(f)\}$ be any number. If $h \in \mathbb{F}[\mathbf{x}]$ is a polynomial such that $\mathcal{H}_{\leq i-1}[f] = \mathcal{H}_{\leq i-1}[h]$, then*

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

Proof. For the rest of the proof, we think of $P(\mathbf{x}, y)$ as an element of $\mathbb{F}[\mathbf{x}][y]$. Henceforth, we drop the variables $\mathbf{x}$ everywhere, and think of P as a univariate in y. Thus, $P(y) = P(\mathbf{x}, y)$. 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, $\mathcal{H}_{\leq i}[P(f)] = \mathcal{H}_{\leq i-1}[P(f)] = 0$. Moreover, since $\mathcal{H}_{\leq i-1}[h] = \mathcal{H}_{\leq i-1}[f]$, we get that $\mathcal{H}_{\leq i-1}[P(f)] = \mathcal{H}_{\leq i-1}[P(h)] = 0$. So, we have

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

We first observe that if $f_i = h_i$, then $\mathcal{H}_{\leq i}[P(h)] = 0$, and the lemma is trivially true. So, for the rest of the argument, we assume that $f_i - h_i \neq 0$. Now, by using Lemma 4.2, we get the following equality.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}\left[P(h) + P^{(1)}(h) \cdot (f_i - h_i) + P^{(2)}(h) \cdot (f_i - h_i)^2/2! + \cdots + P^{(r)}(h) \cdot (f_i - h_i)^r/r!\right] \\ &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_{\leq i}\left[P^{(1)}(h) \cdot (f_i - h_i)\right] + \cdots + \mathcal{H}_{\leq i}\left[P^{(r)}(h) \cdot (f_i - h_i)^r/r!\right]. \end{aligned}$$

[10] Variants of this lemma, often referred to as the Schwartz–Zippel Lemma, or the DeMillo–Lipton–Schwartz–Zippel Lemma, were discovered at least six times, starting with Øystein Ore in 1922 and David Muller in 1954 [30, 25, 36, 6, 42, 37]. For a brief history, see [2] where the term "Polynomial Identity Lemma" is attributed to L. Babai.

Here, r denotes the degree of P. Since $f_i - h_i$ is non-zero, and 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.

$$\begin{aligned} 0 &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_{\leq i}\left[P^{(1)}(h) \cdot (f_i - h_i)\right] \\ &= \mathcal{H}_{\leq i}[P(h)] + \mathcal{H}_0\left[P^{(1)}(h)\right] \cdot (f_i - h_i)\,. \end{aligned}$$

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

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

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

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

For our proof, we shall look at the structure of the outcome of the lifting operation in Lemma 5.1 more closely. Before proceeding further, we need the following crucial lemma.

Lemma 5.2. *Let $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial of degree r, let $\alpha \in \mathbb{F}$ be a field element and $d \in \mathbb{N}$ be a positive integer. Let $\mathcal{G}'_y(P, \alpha, d)$ be the set of polynomials defined as follows.*

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

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

- *For every $g \in \mathcal{G}_y(P, \alpha, d)$, the degree of every non-zero monomial in g is at least 1 and at most d.*
- $|\mathcal{G}_y| \leq d + 1$.
- *If P has a circuit of size s and depth Δ, then every $g \in \mathcal{G}_y(P, \alpha, d)$ has a circuit of size $O(sr^4)$ and depth Δ.*

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

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

Now, for any $j \in \{0,1,2,\dots,d\}$, by Lemma 4.2, $\frac{\partial^j P}{\partial y^j}(\mathbf{x},y)$ is a scalar multiple of the coefficient of z^j in $P(\mathbf{x},y+z)$. Moreover,

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

Thus, for every $j \in \{0,1,\dots,d\}$, the coefficient of z^j in $P(\mathbf{x},y+z)$ is given by $\sum_{i=j}^{r} \binom{i}{j} C_i(\mathbf{x}) \cdot y^{i-j}$. From Lemma 4.9, we know that each $C_i(\mathbf{x})$ has a circuit of depth Δ and size at most $O(sr)$. Thus, we can obtain a circuit for $\binom{i}{j} C_i(\mathbf{x}) \cdot y^{i-j}$ by adding an additional layer of $\times$ gates on top of the circuit for $C_i(\mathbf{x})$. This increases the size by an additive 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 $C_i(\mathbf{x})$, and this does not incur any additional increase in size. Thus, the polynomials $\frac{\partial^j P}{\partial y^j}(\mathbf{x},y)$, and hence $\frac{\partial^j P}{\partial y^j}(\mathbf{x},\alpha)$ have a circuit of size $O(sr^3)$ and depth Δ. To compute the homogeneous components of these polynomials of degree at most d, we use Lemma 4.9. This increases the size by a factor of at most $O(r^2)$ while keeping the depth the same. $\square$

We now state our key technical observation.

Lemma 5.3. *Let $P \in \mathbb{F}[\mathbf{x},y]$ and $f \in \mathbb{F}[\mathbf{x}]$ be polynomials of degree r and d respectively such that $P(\mathbf{x},f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\mathbf{x},f(\mathbf{x}))\right] = \delta \neq 0$. Let the polynomials in the set $\mathcal{G}_y(P,\mathcal{H}_0[f],d)$ be denoted by $g_0,g_1,\dots,g_d$. Then, for every $i \in \{1,2,\dots,d\}$, there is a polynomial $A_i(\mathbf{z})$ in $d+1$ variables such that the following are true.*

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

This is an analog of the main technical lemma in [8], which we state below.

Lemma 5.4 ([8, Lemma 3.1])**.** *Let $P \in \mathbb{F}[\mathbf{x},y]$ and $f \in \mathbb{F}[\mathbf{x}]$ be polynomials of degree r and d respectively such that $P(\mathbf{x},f) = 0$ and $\mathcal{H}_0\left[\frac{\partial P}{\partial y}(\mathbf{x},f(\mathbf{x}))\right] = \delta \neq 0$. Let $P(\mathbf{x},y) = \sum_{i=0}^{k} C_i(\mathbf{x}) \cdot y^i$. Then, for every $i \in \{1,2,\dots,\deg(f)\}$, there is a polynomial $A_i(\mathbf{z})$ in $k+1$ variables such that,*

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

The difference between these lemmas is that in [8], it is shown that there is a set of polynomials of size $\deg_y(P)+1$ 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 $\deg_y(P) \ge n$, Lemma 5.4

does not say anything non-trivial since f can be trivially written as a polynomial in the n original variables. In contrast, Lemma 5.3 continues to say something non-trivial, as long as $d \ll n$, regardless of the value of $\deg_y(P)$. We now proceed with the proof.

Proof of Lemma 5.3. For the rest of the proof, we think of $P(\mathbf{x}, y)$ as an element of $\mathbb{F}[\mathbf{x}][y]$. So, we drop the variables $\mathbf{x}$ everywhere, and think of P as a univariate in y. Thus, $P(y) = P(\mathbf{x}, y)$. For brevity, we denote $\mathcal{H}_j[f]$ by f_j for every $j \in \mathbb{N}$. We also use $\mathcal{G}_y$ for $\mathcal{G}_y(P, f_0, d)$. The proof will be by induction on i and crucially use Lemma 5.1.

- **Base case.** We first prove the lemma for $i = 1$. We invoke Lemma 5.1 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 $\mathcal{H}_1[P(f_0)] = \mathcal{H}_1[g_0]$ where $g_0 = \mathcal{H}_{\leq d}[P(f_0)] - \mathcal{H}_0[P(f_0)] \in \mathcal{G}_y$.

- **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 5.1 with $h = A_{i-1}(g_0, g_1, \ldots, g_d)$, which exists by the induction hypothesis.
$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h - \frac{P(h)}{\delta}\right].$$
Recall that $\mathcal{H}_0(h) = \mathcal{H}_0(f)$. Thus, $h = f_0 + \tilde{h}$, where the constant term of $\tilde{h}$ is 0 and thus every monomial has degree at least 1. By Lemma 4.2,
$$P(f_0 + \tilde{h}) = P(f_0) + P^{(1)}(f_0) \cdot \tilde{h} + \cdots + P^{(r)}(f_0) \cdot \tilde{h}^r / r!\,.$$
Thus, as $\tilde{h}$ has degree at least 1, we have
$$\begin{aligned}\mathcal{H}_{\leq i}[f] &= \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(P(f_0) + P^{(1)}(f_0) \cdot \tilde{h} + \cdots + P^{(r)}(f_0) \cdot \tilde{h}^r / r!\right)\right] \\ &= \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(P(f_0) + P^{(1)}(f_0) \cdot \tilde{h} + \cdots + P^{(i)}(f_0) \cdot \tilde{h}^i / i!\right)\right].\end{aligned}$$
Since we are only interested in $i \leq d$, the following equality is also true.
$$\mathcal{H}_{\leq i}[f] = \mathcal{H}_{\leq i}\left[h - \frac{1}{\delta} \cdot \left(\mathcal{H}_{\leq d}[P(f_0)] + \mathcal{H}_{\leq d}\left[P^{(1)}(f_0)\right] \cdot \tilde{h} + \cdots + \mathcal{H}_{\leq d}\left[P^{(i)}(f_0)\right] \cdot \tilde{h}^i / i!\right)\right].$$
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 $\mathcal{G}$.[11] For every $j \in \{0, 1, 2, \ldots, i\}$, let $\ell_j(\mathbf{z})$ be an affine form such that $\ell_j(g_0, g_1, \ldots, g_d) = \mathcal{H}_{\leq d}\left[P^{(j)}(f_0)\right]$.

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

Now, we define $A_i(\mathbf{z})$ as

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

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

The following is an easy corollary of Lemma 5.3.

Corollary 5.5. *Let $P(\mathbf{x},y) \in \mathbb{F}[\mathbf{x},y]$ be a polynomial of degree, and $\alpha \in \mathbb{F}$ be such that $P(\mathbf{0},\alpha) = 0$, and $\frac{\partial P}{\partial y}(\mathbf{0},\alpha) \neq 0$. Then, for every $k \in \mathbb{N}$, there is a* unique *polynomial $h_k(\mathbf{x})$ such that $\deg(h) \leq k$, $h_k(\mathbf{0}) = \alpha$, and $\mathcal{H}_{\leq k}[P(\mathbf{x},h_k(\mathbf{x}))] = 0$. Moreover, if the polynomials in the set $\mathcal{G}_y(P,\alpha,k)$ be denoted by $g_0,g_1,\ldots,g_k$. Then, there is a polynomial $A_k(\mathbf{z})$ in $k+1$ variables such that the following are true.*

- *$h_k = \mathcal{H}_{\leq k}[A_k(g_0,g_1,\ldots,g_k)]$, and*
- *$A_k(\mathbf{z})$ is computable by a circuit of size $10k^3$.*

The lemma initially starts with an $\alpha \in \mathbb{F}$ such that α is a root of multiplicity 1 of $P(\mathbf{0},y)$. And, starting from this α, we can lift *uniquely* to a polynomial h_i which is an *approximate* root of the polynomial P. This corollary will be useful later on in the paper, when we study the structure of factors of P which are not linear in y. And, the uniqueness will be important for this.

We are now ready to complete the proof of Theorem 2.5.

Proof of Theorem 2.5. The first step is to massage the circuit for P so that the hypothesis of Lemma 5.3 holds. We will have to keep track of the size and depth blowups 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 $P(\mathbf{x},y) = \sum_{i=0}^{r} y^i C_i(\mathbf{x})$. Let $m \geq 1$ be the multiplicity of f as a root of $P(\mathbf{x},y)$. Thus, $\frac{\partial^j P}{\partial y^j}(\mathbf{x},f) = 0$ for $j \in \{0,1,2,\ldots,m-1\}$, but $\frac{\partial^m P}{\partial y^m}(\mathbf{x},f) \neq 0$. The idea is to just work with the polynomial $\tilde{P} = \frac{\partial^{m-1} P}{\partial y^{m-1}}(\mathbf{x},y)$ 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 shallow circuit. This follows from the proof of the third item in Lemma 5.2, where we argued that $\frac{\partial^j P}{\partial y^j}(\mathbf{x},y)$ has a depth-Δ circuit of size $\mathsf{poly}(s,r)$.

Translating the origin. From the step above, we can assume without loss of generality that $\frac{\partial P}{\partial y}(\mathbf{x},f) \neq 0$. Thus, there is a point $\mathbf{a} \in \mathbb{F}^n$ such that $\frac{\partial P}{\partial y}(\mathbf{a},f(\mathbf{a})) \neq 0$. By translating the origin, we will assume that $\frac{\partial P}{\partial y}(\mathbf{0},f(0)) \neq 0$. 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 a factor of at most n.

Degree of A_d. From Lemma 5.3, we know that the polynomial $A_d(\mathbf{z})$ has a circuit of size $O(d^3)$. To obtain a circuit for f, we first prune away all the homogeneous components of $A_d(\mathbf{z})$ of degree larger than d. Recall that by definition, for every polynomial $g_i \in \mathcal{G}_y$, every non-zero monomial in g_i 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 $A_d(\mathbf{z})$ contributes no monomial of degree at most d in the variables $\mathbf{x}$ 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 $A_d(\mathbf{z})$ of degree at most d.

By Theorem 4.8, we know that given a circuit for $A_d(\mathbf{z})$, we can construct a circuit for $\mathcal{H}_i[A_d(\mathbf{z})]$ by increasing the size of the circuit by a multiplicative factor of at most $O(i^2)$. Thus, $\mathcal{H}_{\leq d}[A_d(\mathbf{z})]$ can be computed by a circuit of size $O(d^3) \times \mathrm{size}(A_d(\mathbf{z}))$. Thus, for the rest of this argument, we will assume that $A_d(\mathbf{z})$ has a circuit of size $O(d^6)$ and degree at most d, and

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

Shallow Circuit for $A_d(\mathbf{z})$. Given that $A_d(\mathbf{z})$ has a circuit of size $O(d^6)$ and degree at most d, by Theorem 4.4, we know that $A_d(\mathbf{z})$ can be computed by a $\sum\prod\sum$ circuit Ψ of size at most $d^{O(\sqrt{d})}$. Similarly, by Theorem 4.3, we know that for any $k \in \mathbb{N}$, $A_d(\mathbf{z})$ can be computed by a depth-$(2k)$ circuit of size $d^{O(d^{1/k})}$.

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

- The size of Ψ' is $(srn)^{10} \cdot d^{O(\sqrt{d})})$.
- The depth of Ψ' is at most $\Delta + 3$. This follows by combining the bottom $\sum$ layer of the $\sum\prod\sum$ circuit for $A_d(\mathbf{z})$ with the top $\sum$ layer of the circuits for $g_i \in \mathcal{G}_y$.
- The degree of Ψ' is at most d^2. This is true because the degree of $A_d(\mathbf{z})$ is at most d (as argued earlier in this proof), and the degree of every polynomial in $\mathcal{G}_y$ is at most d (first item in Theorem 5.2).
- $f = \mathcal{H}_{\leq d}[\Psi'(\mathbf{x})]$.

Note that for any $k \in \mathbb{N}$, we can get Ψ' of size $(srn)^{10} \cdot d^{O(d^{1/k})})$ and of depth at most $\Delta + O(k)$.

To obtain a circuit for f, we apply Lemma 4.9 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. $\square$

6 From roots to arbitrary factors

In this section, we show that Theorem 2.5 essentially generalizes to arbitrary factors, and not necessarily factors of the form $y - f(\mathbf{x})$, up to some loss in the size and depth parameters. The techniques for this generalization are quite standard and well known in this literature, and our presentation here follows the approach of Oliveira [29]. We sketch the main steps towards obtaining circuits for arbitrary factors.

Making the polynomial monic in y. Starting with an arbitrary polynomial $P(\mathbf{x},y)$, we first make sure that it is monic in y. We do this by taking an invertible linear transformation $x_i \to x_i + a_i \cdot y$, where the vector $\mathbf{a}$ is chosen randomly from some large enough grid. Indeed, assume that $\deg(P) = r$. Let us consider the homogeneous component of degree r of $P(\mathbf{x},y)$. Since $\mathcal{H}_r[P(\mathbf{x},y)]$ is homogeneous in $(\mathbf{x},y)$ of degree r, so $\mathcal{H}_r[P(\mathbf{x},y)] = P_r(\mathbf{x}/y,1)\cdot y^r$ for a polynomial P_r, implying that $P_r(\mathbf{x}/y,1)$ is not the zero polynomial, so we can write

$$P(\mathbf{x}+\mathbf{a}y,y) = P_r(\mathbf{a},1)y^r + \text{lower order terms (in } y\text{)}\,.$$

By Lemma 4.13, there exists some $\mathbf{a} \in [r+1]^n$, with $P_r(\mathbf{a},1) \neq 0$. Thus, in the inverted coordinate system, the leading coefficient of $P(\mathbf{x}+\mathbf{a}y,y)$ (as a polynomial in y), is some non-zero element of the field $\mathbb{F}$, and, without loss of generality, we can take it to be 1.

If $P(\mathbf{x},y)$ is monic, then so are its factors. To see this, first recall the Gauss Lemma. We shall use it to deduce that the factors of $P(\mathbf{x},y)$ are elements in $\mathbb{F}[\mathbf{x},y]$.

Lemma 6.1 (Gauss Lemma)**.** *Let R be a Unique Factorization Domain with a field of fractions F and let $f(y) \in R[y]$. If $f(y)$ is reducible over $F[y]$, then f is reducible over $R[y]$.*

Now, we have the following simple observation.

Observation 1. Let $R = \mathbb{F}[x]$. If $P \in R[y]$ is a monic polynomial in y, and $P = g\cdot h$, where $g,h \in R[y]$, then the leading coefficients of g and h in y belong to $\mathbb{F}\setminus\{0\}$.

Thus, for the rest of this section, we will assume that all the factors of $P(\mathbf{x},y)$ are also monic in y.

Working over the algebraic closure of $\mathbb{F}(\mathbf{x})$. As above, P is monic in y with $\deg_y(P) = r$, that is,

$$P(\mathbf{x},y) = y^r + \sum_{i=0}^{r-1} P_r(\mathbf{x})y^i.$$

Assume that P does not factor into linear factors in y, and that $f(\mathbf{x},y)$ is one of its factors, of degree k in y. Since P is monic in y, we know that f must also be monic in y. Working over the algebraic closure of $\mathbb{F}(\mathbf{x})$ (that is, the field $\overline{\mathbb{F}(\mathbf{x})}$), we can factor P (and f) into linear factors in y. The algebraic closure of $\mathbb{F}(\mathbf{x})$ is a complicated object, but we only need to think of elements of the closure as "functions" over the variables in $\mathbf{x}$. Since f divides P , if

$$P(\mathbf{x},y) = \prod_{i=1}^{r}(y-\varphi_i(\mathbf{x}))\,,$$

without loss of generality, assume the first d of these φ_i correspond to roots of f, so we have

$$f(\mathbf{x},y) = \prod_{i=1}^{d}(y-\varphi_i(\mathbf{x}))\,.$$

We note that $\varphi_i(\mathbf{x})$ may not be polynomials in $\mathbf{x}$.[12] Still, the fact that they share some roots in the closure of $\mathbb{F}(\mathbf{x})$ gives us a way to approximate them, using Hensel's lifting, similar to Lemma 5.3. For the rest of our argument, we first need to ensure some non-degeneracy conditions.

[12] As shown in [7], $\varphi_i(\mathbf{x})$ could be a power series in $\mathbf{x}$.

Reducing the multiplicity of f in P. We first make sure that f is a factor of multiplicity 1 of P; if f is a factor of multiplicity $m > 1$, we can replace P by $\tilde{P} = \frac{\partial^{m-1} P}{\partial y^{m-1}}(\mathbf{x}, y)$. Clearly, f is a factor of multiplicity exactly 1 of $\tilde{P}$. Ensuring that $\tilde{P}$ can also be computed by a small shallow circuit, follows from the proof of the third item in Lemma 5.2, where we argued that $\frac{\partial^j P}{\partial y^j}(\mathbf{x}, y)$ has a depth-Δ circuit of size $O(sr^3)$. So, for the rest of the proof, we will assume that f is a factor of P of multiplicity equal to 1.

Properly separating shifts. To proceed further, we want a shift in $\mathbf{x}$ such that each factor has no repeating roots in y and distinct factors share no root in y. This follows from the below lemma from [29], which we state without a proof.

Lemma 6.2 ([29, Lemma 3.6]). *Let $f(\mathbf{x}, y), g(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be polynomials such that $\deg_y(f) \geq 1$, $\deg_y(g) \geq 1$, f is irreducible and f does not divide g. Then, there is a $\mathbf{c} \in \overline{\mathbb{F}}^n$ such that*

- *$f(\mathbf{c}, y)$ is a polynomial with exactly $\deg_y(f)$ distinct roots in $\mathbb{F}$, and*
- *$f(\mathbf{c}, y)$ and $g(\mathbf{c}, y)$ have no common roots.*

Now, let us consider the polynomial $g = P/f$. Since f is factor of multiplicity 1 of P, f does not divide g. From Lemma 6.2, we know that there is a $\mathbf{c} \in \mathbb{F}^n$ such that $f(\mathbf{c}, y)$ and $g(\mathbf{c}, y)$ do not share a root, and all the roots of $f(\mathbf{c}, y)$ are distinct. At the cost of increasing the depth of the circuit of P by 1, we can assume without loss of generality that $\mathbf{c}$ is the origin. So, for the rest of the proof, we assume that $f(\mathbf{0}, y)$ has no repeating roots, and $f(\mathbf{0}, y)$ and $g(\mathbf{0}, y)$ share no common roots. Let $\alpha_1, \alpha_2, \ldots, \alpha_r$ be the roots of $P(\mathbf{0}, y)$ and let $\alpha_1, \alpha_2, \ldots, \alpha_d$ be the roots of $f(\mathbf{0}, y)$.

Approximating the roots of P. The goal of this step is to approximate the roots of P by low-degree polynomials with small circuits. From the previous paragraph, we know that for $i \in [d]$, $P(\mathbf{0}, \alpha_i) = 0$ and $\frac{\partial P}{\partial y}(\mathbf{0}, \alpha_i) \neq 0$. Thus, from Corollary 5.5, there are polynomials $q_1, q_2, \ldots, q_d$ of degree at most d such that for every $i \in [d]$, there is a polynomial $A_{i,d}(\mathbf{z})$ in $d+1$ variables such that the following are true.

- $q_i(\mathbf{0}) = \alpha$, and
- $q_i = \mathcal{H}_{\leq d}[A_{i,d}(g_{i,0}, g_{i,1}, \ldots, g_{i,d})]$, and
- $A_{i,d}(\mathbf{z})$ is computable by a circuit of size at most $10d^3$.

Here, for every $i \in [d]$, $g_{i,0}, g_{i,1}, \ldots, g_{i,d}$ are the polynomials in the set $\mathcal{G}_y(P, \alpha_i, d)$. Thus, we have degree d polynomials, which are approximations of the roots of P, the constant terms of these polynomials agree with the roots of $f(\mathbf{x}, 0)$ and these approximate roots have *small* shallow circuits. Moreover, We will now combine these approximations to obtain a circuit for f.

Obtaining a circuit for f. In this final step, we are going to obtain circuit for f from the polynomials $q_1, q_2, \ldots, q_d$ in the previous step. The first observation is that the $q_1, q_2, \ldots, q_d$ are also approximate roots of the polynomial f. To see this, observe that by our choice, $\alpha_1, \alpha_2, \ldots, \alpha_d$ are distinct roots of $f(\mathbf{0}, y)$. Thus, for each $i \in [d]$, $f(\mathbf{0}, \alpha_i) = 0$ and $\frac{\partial f}{\partial y}(\mathbf{0}, \alpha_i) \neq 0$. Thus, by Corollary 5.5, there are degree

d polynomials $\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_d$ of degree at most d such that $\mathcal{H}_{\leq d}[f(\mathbf{x}, \tilde{q}_i(\mathbf{x}))] = 0$. Thus, we also have $\mathcal{H}_{\leq d}[P(\mathbf{x}, \tilde{q}_i(\mathbf{x}))] = 0$. So, by the uniqueness condition in Corollary 5.5, we get that the set of polynomials $\{\tilde{q}_i : i \in [d]\}$ must be the same as $\{q_i : i \in [d]\}$.

Next, to obtain a circuit for f, we now claim that

$$f(\mathbf{x}, y) = \mathcal{H}_{\leq d}\left[\prod_{i=1}^{d}(y - q_i(\mathbf{x}))\right].$$

The proof of this fact follows immediately from Lemma 5.4 in [29]. We state a special case, which suffices for our application.

Lemma 6.3 ([29, Lemma 5.4]). *Let $P(\mathbf{x}, y)$ and $f(\mathbf{x}, y)$ be polynomials of degree r and d respectively, such that P and f are monic in y, f is a factor of P and all the roots of $f(\mathbf{0}, y)$ are distinct and roots of multiplicity exactly one of $P(\mathbf{0}, y)$. Let $\alpha_1, \alpha_2, \ldots, \alpha_d$ be the roots of $f(\mathbf{0}, y)$ and let $q_1, q_2, \ldots, q_d \in \mathbb{F}[\mathbf{x}]$ be polynomials of degree at most d such that for every $i \in [d]$,*

- *$q_i(\mathbf{0}) = \alpha_{\mathbf{i}}$,*
- *$\mathcal{H}_{\leq d}[P(\mathbf{x}, q_i(\mathbf{x}))] = \mathcal{H}_{\leq d}[f(\mathbf{x}, q_i(\mathbf{x}))] = 0$.*

Then,

$$f = \mathcal{H}_{\leq d}\left[\prod_{i=1}^{d}(y - q_i(\mathbf{x}))\right].$$

Thus, given the circuits for $q_i(\mathbf{x})$, we can obtain a circuit for $f(\mathbf{x}, y)$ by increasing the depth by at most two (a product layer, and then a sum layer for interpolation), and size by a $\mathsf{poly}(d)$ factor. In summary, we have the following two statements.

Lemma 6.4. *Let $P \in \mathbb{F}[\mathbf{x}, y]$ and $f \in \mathbb{F}[\mathbf{x}, y]$ be polynomials of degree r and d respectively such that P is monic in y and f is an irreducible factor of P. Then, there exist $\mathbf{c} \in \mathbb{F}^n$, $\alpha_1, \alpha_2, \ldots, \alpha_d \in \mathbb{F}$ and a polynomial $B(\mathbf{z})$ of degree at most d in $t = O(d^2)$ variables, such that the following are true.*

- *$f(\mathbf{x} + \mathbf{c}, y) = \mathcal{H}_{\leq d}\left[B(g_0, g_1, \ldots, g_t)\right]$, where $g_1, g_2, \ldots, g_t$ are polynomials in the set*

$$\bigcup_{i=1}^{d} \mathcal{G}_y\left(P(\mathbf{x} + \mathbf{c}, y), \alpha_i, d\right).$$

- *$B(\mathbf{z})$ is computable by a circuit of size $\mathsf{poly}(d)$.*

Theorem 6.5. *Let $P \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial of degree r in $n+1$ variables that can be computed by an arithmetic circuit of size s of depth Δ. Let $f \in \mathbb{F}[\mathbf{x}, y]$ be an irreducible polynomial of degree d such that f divides P. Then, f can be computed by a circuit of depth $\Delta + O(1)$ and size $\mathsf{poly}(s, r, n) \cdot d^{O(\sqrt{d})}$.*

7 Deterministic PIT for shallow circuits from hardness

In this section, we use Theorem 2.5 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. In short, the high-level strategy is to generate hitting set for shallow circuits from the hard polynomial combined with Nisan designs. Since the content of this part is very similar to the proofs of similar statements in [13] and [8], we only outline the differences in the proofs (if any), and refer the reader to [8] for details. We start with the following lemma, which is the analog of Lemma 4.1 in [8].

Lemma 7.1 (Analog of Lemma 4.1 in [8])**.** *Let $q(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ 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 4.12, so that $\ell = O(m^2/\log n)$, $|S_i| = m$, and $\left|S_i \cap S_j\right| \leq \log n$. For a multilinear polynomial $f \in \mathbb{F}[z_1, z_2, \ldots, z_m]$ of degree d, put*

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

If $Q(\mathbf{y}) \equiv 0$, then $f(\mathbf{z})$ 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 $d \ll m$, while the individual degree of q is allowed to be unbounded, whereas the bound in [8] becomes trivial once $\deg_y(q)$ is larger than m.

Proof Sketch. The proof is along the lines of the proof of Lemma 4.1 in [8]. We now give a sketch of the details. We first define the hybrid polynomials $Q_0(\mathbf{x},\mathbf{y}), Q_1(\mathbf{x},\mathbf{y}), \ldots, Q_n(\mathbf{x},\mathbf{y})$ as follows.

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

We know that $Q_0(\mathbf{x},\mathbf{y})$ is non-zero, whereas $Q_n(\mathbf{x},\mathbf{y})$ is identically zero. Thus, there is an $i \in \{0,1,\ldots,n\}$ such that $Q_i(\mathbf{x},\mathbf{y}) \not\equiv 0$ and $Q_{i+1}(\mathbf{x},\mathbf{y}) \equiv 0$. We now fix the variables $x_{i+2}, x_{i+3}, \ldots, x_n$ and the variables $\left\{y_j : j \notin S_{i+1}\right\}$ to field constants while maintaining the non-zeroness of Q_i. This can be done via Lemma 4.13. Thus, we have a polynomial $\tilde{q}$ by fixing the aforementioned variables such that the following two conditions hold.

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

Let $A_0(\mathbf{y}|_{S_{i+1}}, x_{i+1})$ denote the polynomial $\tilde{q}\left(f(\mathbf{y}|_{S_1\cap S_{i+1}}), f(\mathbf{y}|_{S_2\cap S_{i+1}}), \ldots, f(\mathbf{y}|_{S_i\cap S_{i+1}}), x_{i+1}\right)$. The above two conditions imply that $f(\mathbf{y}|_{S_{i+1}})$ is a root of the polynomial $A_0(\mathbf{y}|_{S_{i+1}}, x_{i+1}) \in \mathbb{F}[\mathbf{y}|_{S_{i+1}}][x_{i+1}]$, viewed as a polynomial in x_{i+1}. Moreover, $A_0(\mathbf{y}|_{S_{i+1}}, x_{i+1})$ has a circuit of size $O(sn)$ and depth at most $\Delta+2$. This follows from the fact that $f(\mathbf{y}|_{S_1\cap S_{i+1}})$ 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 2.5 implies that $f(\mathbf{y}|_{S_{i+1}})$ can be computed by a circuit of size $O(\mathsf{poly}(s,n,D)d^{O(\sqrt{d})})$ $\mathsf{poly}(s,n,D)d^{O(\sqrt{d})}$ and depth at most $\Delta+5$, thus completing the proof. ∎

We now sketch the proof of Theorem 2.3.

Proof Sketch. Once again, the proof follows the proof of Theorems 1 and 2 in [8]. Let $\{f_m\}$ be an explicit family of 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 $C \in \mathbb{F}[\mathbf{x}]$ of size s, depth $\Delta - 5$ and degree D on n variables, let $q \in \mathbb{F}[\mathbf{x}]$ be the polynomial computed by C. The goal here is to determine whether q is non-zero. From the equivalence of black-box PIT and hitting set, it suffices to construct hitting sets for the circuit class with the above properties.

- We construct a design $S_1, S_2, \ldots, S_n \subseteq [\ell]$ using Theorem 4.12 where each set S_i has size m, $\ell = O(m^2/\log n) \leq n^{2\varepsilon} < n^{0.98}$ and $\left|S_i \cap S_j\right| \leq \log n$. 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

  $$q\left(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\right)$$

  on all points of T^{ℓ}.

  $$H = \{(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})) \mid \mathbf{y} \in T^{\ell}\}$$

  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 $q\left(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\right)$ is not identically zero, then it has degree at most Dd and hence by Lemma 4.13, q does not vanish on the set H. Else, $q\left(f(\mathbf{y}|_{S_1}), f(\mathbf{y}|_{S_2}), \ldots, f(\mathbf{y}|_{S_n})\right) \equiv 0$. But then, by Lemma 7.1, we get that f can be computed by a circuit of depth Δ and size $\mathsf{poly}(s,n,D)d^{O(\sqrt{d})}$. 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. $\square$

8 Factors of polynomials in VNP

We now prove Theorem 2.9, which is restated below.

Theorem 8.1 (Theorem 2.9 restated)**.** *Let $P(\mathbf{x})$ be a polynomial of degree r over $\mathbb{F}$, and let $Q(\mathbf{x},\mathbf{y})$ be a polynomial in $n+m$ variables such that*

$$P(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^m} Q(\mathbf{x},\mathbf{y}),$$

and Q can be computed by a circuit of size s. Let f be an irreducible factor of P of degree d. Then, there exists an $m' \leq \mathsf{poly}(s,r,d,n,m)$ and polynomial $h(\mathbf{x}, z_1, z_2, \dots, z_{m'})$, such that $h(\mathbf{x}, \mathbf{z})$ can be computed by a circuit of size $s' \leq \mathsf{poly}(s,r,d,n,m)$ and

$$f(\mathbf{x}) = \sum_{\mathbf{z} \in \{0,1\}^{m'}} h(\mathbf{x}, \mathbf{z}) .$$

For our proof, we use the following structure theorem of Valiant [40], and its consequences (Claim 8.4). Below, we state the theorem, and then use it to prove Theorem 8.1. For completeness, we include a proof using the depth-reduction results in [41] in the appendix.

Theorem 8.2 (Valiant [40])**.** *Let $P(\mathbf{x})$ be a homogeneous polynomial of degree r in n variables that can be computed by an arithmetic circuit C of size s. Then, there is an $m \leq \mathsf{poly}(s,r)$ and a polynomial $Q(\mathbf{x}, y_1, y_2, \dots, y_m)$ such that*

$$P(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{m}} Q(\mathbf{x}, \mathbf{y}) ,$$

and $Q(\mathbf{x}, \mathbf{y})$ can be computed by an arithmetic formula of size $\mathsf{poly}(s,r)$.

We now proceed with the proof of Theorem 8.1.

Proof of Theorem 8.1. Without loss of generality, we will assume that P is monic in a variable z. This can be guaranteed by doing a linear transformation by replacing every variable x_i by $x_i + a_i z$, where a_i are chosen from a large enough grid, based on the degree of P. Note that this preserves the form of P in the hypothesis of the theorem. Moreover, using Theorem 4.8, we will assume that the degree of $Q(\mathbf{x}, \mathbf{y})$ in the variables x and z is r, up to a polynomial blowup in the circuit size of Q.

From Lemma 6.4, we know that there is a $\mathbf{c} \in \mathbb{F}^n$ and a polynomial B in at most $t = O(d^2)$ variables, and polynomials $g_1, g_2, \dots, g_t$ such that

$$f(\mathbf{x} + \mathbf{c}, z) = \mathcal{H}_{\leq d}\left[B(g_1, g_2, \dots, g_t)\right] .$$

For the rest of this proof, we assume that we have shifted the origin, so that $\mathbf{c} = \mathbf{0}$. Again, this just requires replacing every variable x_i by $x_i + c_i$, and this shift of coordinates does not affect the structure of P in the hypothesis of the theorem. Thus,

$$f(\mathbf{x}, z) = \mathcal{H}_{\leq d}\left[B(g_1, g_2, \dots, g_t)\right] .$$

Moreover, B has a circuit of size $\mathsf{poly}(d)$ and each g_i belongs to some set $\mathcal{G}_z(P, \alpha, d)$ for some $\alpha \in \mathbb{F}$. We now need the following two structural claims which follow from direct applications of properties of polynomials in VNP as shown by Valiant [40].

Claim 8.3 (Valiant [40])**.** *For every choice of $\alpha \in \mathbb{F}$ and $g_j \in \mathcal{G}_z(P, \alpha, k)$, there is a polynomial*

$$Q'_j(\mathbf{x}, y_1, y_2, \dots, y_m)$$

such that

$$g_j(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{m}} Q'_j(\mathbf{x}, \mathbf{y}) .$$

Moreover, Q' can be computed by a circuit of size $\mathsf{poly}(s,r,d)$.

The second claim is about the structure of the composed polynomial $B(g_1, g_2, \ldots, g_t)$. This is a special case of a more general result of Valiant [40], which showed that VNP is closed under *composition*.

Claim 8.4 (Valiant [40]). *There is an* $\tilde{m} \leq \mathsf{poly}(m,d)$ *and a polynomial* $\tilde{Q}(\mathbf{x}, y_1, y_2, \ldots, y_{\tilde{m}})$ *such that*

$$B(g_1, g_2, \ldots, g_t) = \sum_{\mathbf{y} \in \{0,1\}^{\tilde{m}}} \tilde{Q}(\mathbf{x}, \mathbf{y}) .$$

Moreover, $\tilde{Q}$ *can be computed by a circuit of size* $\mathsf{poly}(s, r, d, n, m)$.

For completeness, we provide a sketch of the proofs of the claims and that of Theorem 8.2 to the appendix. We now use the claims above to complete the proof of Theorem 8.1.

Observe that if we view $\tilde{Q}$ as a polynomial in $\mathbf{x}$ variables with coefficients coming from $\mathbb{F}[\mathbf{y}]$, then, for every $k \in \mathbb{N}$, it follows that

$$\mathcal{H}_k \left[B(g_1, g_2, \ldots, g_t) \right] = \sum_{\mathbf{y} \in \{0,1\}^{\tilde{m}}} \mathcal{H}_{k,\mathbf{x}} \left[\tilde{Q}(\mathbf{x}, \mathbf{y}) \right] .$$

Here, $\mathcal{H}_{k,\mathbf{x}}[\tilde{Q}(\mathbf{x}, \mathbf{y})]$ denotes the homogeneous component of degree k of $\tilde{Q}(\mathbf{x}, \mathbf{y})$ when viewing $\tilde{Q}(\mathbf{x}, \mathbf{y})$ as a polynomial in x variables. It follows from Theorem 4.8, that by blowing up the size of the circuit for $\tilde{Q}$ by a factor of $O(k^2)$, we can obtain a circuit which computes $\mathcal{H}_{k,\mathbf{x}}[\tilde{Q}(\mathbf{x}, \mathbf{y})]$, and this does not affect the y variables in any way. This gives us a representation of $f(\mathbf{x}, z)$ as

$$f(\mathbf{x}) = \sum_{\mathbf{z} \in \{0,1\}^{m'}} h(\mathbf{x}, \mathbf{z})$$

where $m' = \tilde{m} \leq \mathsf{poly}(m,d)$, and h can be computed by a circuit of size $\mathsf{poly}(s, r, d, n, m)$. This completes the proof of the theorem. □

9 Factors of polynomials with small formulas

In this section, we prove the following theorem, which gives an upper bound on the formula complexity of factors of polynomials which have small formulas. We note that this result is not new and was also proved by Dutta et al. in [7]. Since the proof essentially follows from our techniques developed so far and our proof is different from the proof in [7], we include the statement and a proof sketch.

Theorem 9.1 ([7]). *Let* $P(\mathbf{x})$ *be a polynomial of degree* r *in* n *variables which can be computed by an arithmetic formula of size* s*, and let* $f(\mathbf{x})$ *be a factor of* P *of degree* d*. Then,* $f(\mathbf{x})$ *can be computed by an arithmetic formula of size* $\mathsf{poly}(s, r, n, d^{O(\log d)})$.

Proof. The proof is again along the lines of the proof of Theorem 8.1. We first observe that the polynomials in $\mathcal{G}_y(P, \alpha, k)$ have small formulas. This just follows from the proof of Item 3 in Lemma 5.2 and Lemma 4.10.

Now, recall that from Lemma 6.4, we know that the B is a polynomial in at most $O(d^2)$ variables, and can be computed by a circuit of size $\mathsf{poly}(d)$. Thus, by Theorem 4.6, we get that B can be computed by a formula Φ of size at most $d^{O(\log d)}$. Composing Φ with the formulas for the polynomials in $\mathcal{G}_y(P,\alpha,k)$, we get a formula for $B(g_1,g_2,\ldots,g_t)$ of size $\mathsf{poly}(r,s,m,n,d^{O(\log d)})$, and also,

$$f=\mathcal{H}_{\leq d}\left[B(g_1,g_2,\ldots,g_t)\right].$$

All we need now to complete the proof, is a formula for $\mathcal{H}_{\leq d}\left[B(g_1,g_2,\ldots,g_t)\right]$, and this follows from Lemma 4.10. $\square$

We remark that the proof extends to the model of algebraic branching programs. More precisely, the following statement is true.

Theorem 9.2. *Let $P(\mathbf{x})$ be a polynomial of degree r in n variables which can be computed by an algebraic branching program of size s, and let $f(\mathbf{x})$ be a factor of P of degree d. Then, $f(\mathbf{x})$ can be computed by an algebraic branching program of size* $\mathsf{poly}(s,r,n,d^{O(\log d)})$.

10 Proofs of claims

We now include the proofs of Theorem 8.2, Claim 8.3 and Claim 8.4. We follow the notation set up in the proof of Theorem 8.1.

Proof of Claim 8.3. We relabel one of the variables in $\mathbf{x}$ as z. Let $C_0(\mathbf{x}),C_1(\mathbf{x}),\ldots,C_r(\mathbf{x})$ be polynomials such that

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

Recall that $\frac{\partial^j P}{\partial z^j}(\mathbf{x},z)$ equals $j!\cdot\sum_{i=j}^{r}\binom{i}{j}C_i(\mathbf{x})\cdot z^{i-j}$. Now, we know that

$$P(\mathbf{x},z)=\sum_{y\in\{0,1\}^m}Q(\mathbf{x},\mathbf{y},z).$$

Expressing $Q(\mathbf{x},\mathbf{y},z)$ as a univariate in z, we get

$$Q(\mathbf{x},\mathbf{y},z)=\sum_{i=1}^{r}C'_i(\mathbf{x},\mathbf{y})\cdot z^i.$$

Recall that $Q(\mathbf{x},\mathbf{y},z)$ has a circuit of size $\mathsf{poly}(s)$ and degree at most r. By viewing Q as a univariate in z and applying Theorem 4.8, we get that each $C'_i(\mathbf{x},\mathbf{y})$ has a circuit of size $\mathsf{poly}(s,r)$. In particular, for every $j\in\mathbb{N}$, we can write $C_j(\mathbf{x})$ as

$$C_j(\mathbf{x})=\sum_{\mathbf{y}\in\{0,1\}^m}C'_j(\mathbf{x},\mathbf{y}).$$

Therefore, for every $j\in\{0,1,2,\ldots,d\}$, we get

$$\sum_{i=j}^{r}\binom{i}{j}C_i(\mathbf{x})\cdot z^{i-j}=\sum_{\mathbf{y}\in\{0,1\}^m}\left(\sum_{i=j}^{r}\binom{i}{j}C'_i(\mathbf{x},\mathbf{y})\cdot z^{i-j}\right).$$

Moreover, the polynomial $\left(\sum_{i=j}^{r}\binom{i}{j}C'_i(\mathbf{x},\mathbf{y})\cdot z^{i-j}\right)$ has a circuit of size $\mathsf{poly}(n,r)$. This completes the proof of the claim. $\square$

Proof of Claim 8.4. The proof is in two parts. We first define the construction of the circuit for $\tilde{Q}$, and then argue the correctness of this construction.

Constructing $\tilde{Q}$. We know that $B(z_1, z_2, \ldots, z_t)$ is of degree at most d and can be computed by a circuit of size $\mathsf{poly}(d)$. It follows from Theorem 8.2, that there is an $a \le \mathsf{poly}(t,d)$ and a polynomial B' in at most $t+a$ variables such that

$$B(z_1, z_2, \ldots, z_t) = \sum_{\mathbf{y} \in \{0,1\}^a} B'(\mathbf{z}, \mathbf{y}).$$

Crucially, it is also the case that B' has a *formula* of size $\mathsf{poly}(d,t)$. We remark that it is extremely important for the proof that B' has a small formula, and not just a small circuit. To construct $\tilde{Q}$, we consider the formula Φ for $B'(\mathbf{z}, \mathbf{y})$ and let $\ell_1, \ell_2, \ldots, \ell_u$ be the input gates of Φ. Each of these input gates is labeled by a z variable, a y variable or a field constant. From this, we construct a circuit Φ' by going through over the input gates, and replacing the gate ℓ_i by the circuit for polynomial $Q'_j(\mathbf{x}, \mathbf{y}_i)$ from Claim 8.3 if it is labeled by z_j and leaving it unchanged otherwise. Thus, Φ' computes polynomial in variables $\mathbf{x} \cup \mathbf{y} \cup (\bigcup_{j=1}^{u} \mathbf{y}_j)$, of size $\mathsf{poly}(s,d,r,n,m)$. We denote this polynomial by $\tilde{Q}$. Let $\tilde{m} = |\mathbf{x} \cup \mathbf{y} \cup (\bigcup_{j=1}^{u} \mathbf{y}_j)| \le \mathsf{poly}(m,d)$. We now argue that the construction in correct.

Correctness. We now argue that

$$B(g_1, g_2, \ldots, g_t) = \sum_{(\mathbf{y}, \mathbf{y}_1, \ldots, \mathbf{y}_u) \in \{0,1\}^{\tilde{m}}} \Phi'(\mathbf{x}, \mathbf{y}, \mathbf{y}_1, \ldots, \mathbf{y}_u).$$

The proof is by an induction on the size of formula Φ and the fact that in going from Φ to Φ', each of the input gates of Φ which was labeled by a z_j variable was replaced by a copy of Q'_j with a unique copy of the auxiliary y variables. Note that the uniqueness of the auxiliary variables is due to the fact that B' has a formula. Finally, the proof follows from the following observation showing that $\tilde{m} = \mathsf{poly}(s,t,d)$. We skip the details.

Observation 1. Let $R_1(\mathbf{x}), R_2(\mathbf{x})$ and $S_1(\mathbf{x}, \mathbf{y}), S_2(\mathbf{x}, \mathbf{z})$ be polynomials such that

$$R_1(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{|\mathbf{y}|}} S_1(\mathbf{x}, \mathbf{y}), \quad \text{and}$$
$$R_2(\mathbf{x}) = \sum_{\mathbf{z} \in \{0,1\}^{|\mathbf{z}|}} S_2(\mathbf{x}, \mathbf{z}).$$

Then,

$$R_1(\mathbf{x}) + R_2(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{|\mathbf{y}|}, \mathbf{z} \in \{0,1\}^{|\mathbf{z}|}} \left(S_1(\mathbf{x}, \mathbf{y}) + S_2(\mathbf{x}, \mathbf{z})\right), \quad \text{and}$$
$$R_1(\mathbf{x}) \times R_2(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{|\mathbf{y}|}, \mathbf{z} \in \{0,1\}^{|\mathbf{z}|}} \left(S_1(\mathbf{x}, \mathbf{y}) \times S_2(\mathbf{x}, \mathbf{z})\right). \qquad \square$$

Proof of Theorem 8.2. Let C' be the circuit obtained by applying Theorem 4.5 to the circuit C. The idea is to inductively turn C' into a formula while reducing the depth by half in every step. From the properties of C', we get that

$$P = \sum_{i=1}^{s'} A_{i,1} \cdot A_{i,2} \cdots A_{i,5} \,.$$

Here $s' \leq \mathsf{poly}(s,n,d)$ is the size of C', and every $A_{i,j}$ is a polynomial computed by a subcircuit in C' and the degree of $A_{i,j}$ is at most $d/2+1$. We introduce variables $\left\{y_{i,j}, i \in [s'], j \in [5]\right\}$. Let $R(\mathbf{y})$ be the following polynomial.

$$R(\mathbf{y}) = \sum_{i=1}^{s'} (y_{i,1} \cdot y_{i,2} \cdots y_{i,5}) \cdot \prod_{i' \neq i} \left((1-y_{i',1})(1-y_{i',2}) \cdots (1-y_{i',5})\right)$$

Observe that for $\mathbf{b} \in \{0,1\}^{|\mathbf{y}|}$, $R(\mathbf{b})$ is 1 if and only if there is an $i \in [s']$ such that $(b_{i,1}, b_{i,2}, b_{i,3}, b_{i,4}, b_{i,5})$ equals $(1,1,1,1,1)$ and for all $i' \in [s']$ with $i \neq i'$, $(b_{i',1}, b_{i',2}, b_{i',3}, b_{i',4}, b_{i',5}) = (0,0,0,0,0)$, and zero otherwise. Moreover, $R(\mathbf{y})$ can be computed by an arithmetic formula of size $s'^2 = \mathsf{poly}(s)$. Now, observe that we can write the polynomial P as follows.

$$P(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^{5s'}} R(\mathbf{y}) \cdot \prod_{j=1}^{5} \left(\sum_{i=1}^{s'} A_{i,j} y_{i,j} \right) .$$

Also, for every j, the polynomial $\sum_{i=1}^{s'} A_{i,j} y_{i,j}$ is of degree at most $d/2+1$ and can be computed by a circuit of size at most $3s'$. This is true since each $A_{i,j}$ is computed by a subcircuit of C'. Thus, we have expressed a degree d polynomial, computable by a circuit of size s' in terms of polynomials of degree at most $d/2+1$, and circuit complexity $3s'$. We have also had to incur an additional additive cost of $O(s'^2)$ for the formula computing R. The idea of the proof is to keep applying this reduction for $\log d$ iterations, such that the degree of each of the polynomials is at most a constant. Then, we compute these *generating* polynomials by a formula by brute force.

We now argue that the number of y variables introduced in the process, and the total size of the formula for the final verifier is still polynomially bounded in s, d. The number of auxiliary y variables introduced is given by the following recurrence.

$$m(d, s') \leq 5s' + 5m(d/2+1, 3s') \,.$$

The size of the formula $F(d,s)$ is upper bounded by the following recurrence.

$$F(d, s') \leq c \cdot s'^2 + 5F(d/2+1, 3s') \,,$$

where $c > 0$ is some constant. It is not hard to see that both $m(d,s')$ and $F(d,s')$ are upper bounded by a fixed polynomial function of d, s'. □

Acknowledgment

We thank Rafael Oliveira for making us aware of the question about the complexity of factors of polynomials in VNP, and Guy Moshkovitz for helpful discussions.

We also thank the anonymous reviewers at ToC and the editor-in-chief László Babai for various comments and suggestions which helped improve the readability of this paper.

References

[1] MANINDRA AGRAWAL AND V. VINAY: Arithmetic circuits: A chasm at depth four. In *Proc. 49th FOCS*, pp. 67–75. IEEE Comp. Soc. Press, 2008. [doi:10.1109/FOCS.2008.32] 11

[2] VIKRAMAN ARVIND, PUSHKAR S. JOGLEKAR, PARTHA MUKHOPADHYAY, AND S. RAJA: Randomized polynomial-time identity testing for noncommutative circuits. *Theory of Computing*, 15(7):1–36, 2019. [doi:10.4086/toc.2019.v015a007] 14

[3] WALTER BAUR AND VOLKER STRASSEN: The complexity of partial derivatives. *Theoret. Comput. Sci.*, 22(3):317–330, 1983. [doi:10.1016/0304-3975(83)90110-X] 3

[4] PETER BÜRGISSER: *Completeness and Reduction in Algebraic Complexity Theory*. Springer, 2000. [doi:10.1007/978-3-662-04179-6] 7

[5] PETER BÜRGISSER: The complexity of factors of multivariate polynomials. *Found. Computational Math.*, 4(4):369–396, 2004. Preliminary version in FOCS'01. [doi:10.1007/s10208-002-0059-5] 8

[6] RICHARD A. DEMILLO AND RICHARD J. LIPTON: A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193–195, 1978. [doi:10.1016/0020-0190(78)90067-4] 14

[7] PRANJAL DUTTA, NITIN SAXENA, AND AMIT SINHABABU: Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proc. 50th STOC*, pp. 1152–1165. ACM Press, 2018. [doi:10.1145/3188745.3188760, arXiv:1710.03214] 2, 8, 10, 20, 26

[8] ZEEV DVIR, AMIR SHPILKA, AND AMIR YEHUDAYOFF: Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2010. Preliminary version in STOC'08. [doi:10.1137/080735850] 2, 4, 5, 8, 10, 12, 14, 16, 23, 24

[9] MICHAEL A. FORBES: Deterministic divisibility testing via shifted partial derivatives. In *Proc. 56th FOCS*, pp. 451–465. IEEE Comp. Soc. Press, 2015. [doi:10.1109/FOCS.2015.35] 7

[10] HERVÉ FOURNIER, NUTAN LIMAYE, GUILLAUME MALOD, AND SRIKANTH SRINIVASAN: Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015. Preliminary version in STOC'14. [doi:10.1137/140990280] 3, 6

[11] ANKIT GUPTA, PRITISH KAMATH, NEERAJ KAYAL, AND RAMPRASAD SAPTHARISHI: Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, 2014. Preliminary version in CCC'13. [doi:10.1145/2629541] 3, 6

[12] ANKIT GUPTA, PRITISH KAMATH, NEERAJ KAYAL, AND RAMPRASAD SAPTHARISHI: Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Preliminary version in FOCS'13. [doi:10.1137/140957123] 9, 11

[13] VALENTINE KABANETS AND RUSSELL IMPAGLIAZZO: Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complexity*, 13(1-2):1–46, 2004. Preliminary version in STOC'03. [doi:10.1007/s00037-004-0182-6] 2, 3, 4, 5, 23

[14] KYRIAKOS KALORKOTI: A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985. Preliminary version in ICALP'82. [doi:10.1137/0214050] 3

[15] ERICH KALTOFEN: Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985. [doi:10.1137/0214035] 2

[16] ERICH KALTOFEN: Uniform closure properties of P-computable functions. In *Proc. 18th STOC*, pp. 330–337. ACM Press, 1986. [doi:10.1145/12130.12163] 2

[17] ERICH KALTOFEN: Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proc. 19th STOC*, pp. 443–452. ACM Press, 1987. [doi:10.1145/28395.28443] 2

[18] ERICH KALTOFEN: Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, pp. 375–412. JAI Press, 1989. 2, 3, 9

[19] NEERAJ KAYAL, CHANDAN SAHA, AND RAMPRASAD SAPTHARISHI: A super-polynomial lower bound for regular arithmetic formulas. In *Proc. 46th STOC*, pp. 146–153. ACM Press, 2014. [doi:10.1145/2591796.2591847] 6

[20] PASCAL KOIRAN: Arithmetic circuits: The chasm at depth four gets wider. *Theoret. Comput. Sci.*, 448:56–65, 2012. [doi:10.1016/j.tcs.2012.03.041, arXiv:1006.4700] 11

[21] MRINAL KUMAR: A quadratic lower bound for homogeneous algebraic branching programs. *Comput. Complexity*, 28(3):409–435, 2019. Preliminary version in CCC'17. [doi:10.1007/s00037-019-00186-3] 3

[22] MRINAL KUMAR AND RAMPRASAD SAPTHARISHI: An exponential lower bound for homogeneous depth-5 circuits over finite fields. In *Proc. 32nd Computational Complexity Conf. (CCC'17)*, volume 79, pp. 31:1–31:30. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. [doi:10.4230/LIPIcs.CCC.2017.31, arXiv:1507.00177] 6

[23] MRINAL KUMAR AND SHUBHANGI SARAF: On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.*, 46(1):336–387, 2017. Preliminary version in FOCS'14. [doi:10.1137/140999335, arXiv:1404.1950] 3, 6

[24] RUDOLF LIDL AND HARALD NIEDERREITER: *Finite Fields*. Encyclopedia of Mathematics and its Applications. Cambridge Univ. Press, 2nd edition, 1996. [doi:10.1017/CBO9780511525926] 32

[25] DAVID E. MULLER: Application of boolean algebra to switching circuit design and to error detection. *Trans. Inst. Radio Engineers Professional Group on Electronic Computers*, EC-3:6–12, 1954. [doi:10.1109/IREPGELC.1954.6499441] 14

[26] NOAM NISAN: Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991. Preliminary version in FOCS'88. [doi:10.1007/BF01375474] 3, 13

[27] NOAM NISAN AND AVI WIGDERSON: Hardness vs randomness. *J. Comput. System Sci.*, 49(2):149–167, 1994. Preliminary version in FOCS'88. [doi:10.1016/S0022-0000(05)80043-1] 3

[28] NOAM NISAN AND AVI WIGDERSON: Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complexity*, 6(3):217–234, 1996. Preliminary version in FOCS'95. [doi:10.1007/BF01294256] 3, 6

[29] RAFAEL OLIVEIRA: Factors of low individual degree polynomials. *Comput. Complexity*, 25(2):507–561, 2016. [doi:10.1007/s00037-016-0130-2] 2, 4, 8, 9, 10, 19, 21, 22

[30] ØYSTEIN ORE: Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter Ser. I*, 7(15):27, 1922. Polynomial Identity Lemma cited with full proof in [24, Theorem 6.13]. 14

[31] RAN RAZ: Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. Preliminary version in FOCS'04. [doi:10.4086/toc.2006.v002a006] 3, 6

[32] RAN RAZ: Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(7):135–177, 2010. Preliminary version in STOC'08. [doi:10.4086/toc.2010.v006a007] 3, 6

[33] RAN RAZ: Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. Preliminary version in STOC'10. [doi:10.1145/2535928] 6

[34] 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. Preliminary version in FOCS'07. [doi:10.1137/070707932] 3

[35] RAN RAZ AND AMIR YEHUDAYOFF: Lower bounds and separations for constant depth multilinear circuits. *Comput. Complexity*, 18(2):171–207, 2009. Preliminary version in CCC'08. [doi:10.1007/s00037-009-0270-8] 3, 6

[36] WOLFGANG M. SCHMIDT: *Equations over Finite Fields: An Elementary Approach*. Volume 536 of *Lecture Notes in Math.* Springer, 1st edition, 1976. 14

[37] JACOB T. SCHWARTZ: Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, 1980. Preliminary version in EUROSAM'79. [doi:10.1145/322217.322225] 14

[38] AMIR SHPILKA AND AMIR YEHUDAYOFF: Arithmetic circuits: A survey of recent results and open questions. *Found. and Trends Theor. Comput. Sci.*, 5:207–388, 2010. [doi:10.1561/0400000039] 4

[39] SÉBASTIEN TAVENAS: Improved bounds for reduction to depth 4 and depth 3. *Inform. and Comput.*, 240:2–11, 2015. Preliminary version in MFCS'13. [doi:10.1016/j.ic.2014.09.004, arXiv:1304.5777] 11

[40] LESLIE G. VALIANT: Reducibility by algebraic projections. In *Logic and Algorithmic*, volume 28 of *L'Enseignement Mathématique*, pp. 253–268. 1982. 9, 25, 26

[41] LESLIE G. VALIANT, SVEN SKYUM, STUART J. BERKOWITZ, AND CHARLES RACKOFF: Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in MFCS'81. [doi:10.1137/0212043] 10, 12, 25

[42] RICHARD E. ZIPPEL: Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Comput. (EUROSAM'79)*, volume 72 of *LNCS*, pp. 216–226. Springer, 1979. [doi:10.1007/3-540-09519-5_73] 14

AUTHORS

Chi-Ning Chou
Ph. D. student
School of Engineering and Applied Sciences
Harvard University
Cambridge, Massachusetts, USA
cnchou@g.harvard.edu
`http://cnchou.github.io/`

Mrinal Kumar
Postdoctoral fellow
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
mrinalkumar08@gmail.com
`http://mrinalkr.bitbucket.io/`

Noam Solomon
Postdoctoral researcher
Department of Mathematics
Massachusetts Institute of Technology
Cambridge, Massachusetts, USA
noam.solom@gmail.com
https://sites.google.com/site/noamsolomonswebpage/home/

ABOUT THE AUTHORS

CHI-NING CHOU is a Ph. D. student in the Theory of Computation group at Harvard University, where he is advised by Boaz Barak. He received a B. S. from National Taiwan University in 2016 and worked as a research assistant in Academia Sinica for one year, mentored by Kai-Min Chung, who not only showed him how to be an independent researcher but also gave him lots of freedom to explore. His research interests include computational complexity, algebraic complexity, cryptography, quantum computing, and their intersections with other fields. Outside academics, he loves playing baseball, playing Go, listening to classical music, and cooking.

MRINAL KUMAR is a postdoc in the Theory group at the University of Toronto, where his host is Ben Rossman. Prior to this, he was a research fellow in the Program on Lower Bounds in Computational Complexity at the Simons Institute for the Theory of Computing, Berkeley, CA, and a postdoc in the Combinatorics and Complexity Program at the Center for Mathematical Sciences and Applications at Harvard. He received his Ph. D. in Computer Science in May 2017 from Rutgers University where he was advised by Swastik Kopparty and Shubhangi Saraf. His research interests are in arithmetic and Boolean circuit complexity and error correcting codes. Mrinal spent his undergrad years at IIT Madras and owes his interest in Complexity Theory to a delightful class on the topic taught by Jayalal Sarma. Apart from theory, he finds great joy in test cricket and in the adventures of Calvin & Hobbes.

NOAM SOLOMON is a postdoc at the Mathematics department at MIT, and before that he was a postdoc at the Center for Mathematical Sciences and Applications at Harvard University. He holds a Ph. D. in Computer-Science from Tel-Aviv University and a Ph. D. in Number Theory from Ben-Gurion University. His research interests include combinatorial and computational geometry, combinatorics, algebraic complexity and number theory. Outside research, he enjoys sports, music, books, a good movie once in a while, and socializing.