Deterministic factorization of constant-depth algebraic circuits in subexponential time

Somnath Bhattacharjee [*]   Mrinal Kumar [†]   Varun Ramanathan[†]
Ramprasad Saptharishi[†]   Shubhangi Saraf [‡]

Abstract

While efficient randomized algorithms for factorization of polynomials given by algebraic circuits have been known for decades, obtaining an even *slightly non-trivial* deterministic algorithm for this problem has remained an open question of great interest. This is true even when the input algebraic circuit has additional structure, for instance, when it is a constant-depth circuit. Indeed, no efficient deterministic algorithms are known even for the seemingly easier problem of factoring sparse polynomials or even the problem of testing the irreducibility of sparse polynomials.

In this work, we make progress on these questions: we design a deterministic algorithm that runs in subexponential time, and when given as input a constant-depth algebraic circuit C over the field of rational numbers, it outputs algebraic circuits (of potentially unbounded depth) for all the irreducible factors of C, together with their multiplicities. In particular, we give the first subexponential time deterministic algorithm for factoring sparse polynomials.

For our proofs, we rely on a finer understanding of the structure of power series roots of constant-depth circuits and the analysis of the Kabanets-Impagliazzo generator. In particular, we show that the Kabanets-Impagliazzo generator constructed using low-degree hard polynomials (explicitly constructed in the work of Limaye, Srinivasan & Tavenas) preserves not only the non-zeroness of small constant-depth circuits (as shown by Chou, Kumar & Solomon), but also their irreducibility and the irreducibility of their factors.

---

[*]University of Toronto, Canada. Email: `somnath.bhattacharjee@mail.utoronto.ca`

[†]Tata Institute of Fundamental Research, Mumbai, India. Email: `{mrinal, varun.ramanathan, ramprasad}@tifr.res.in`. Research supported by the Department of Atomic Energy, Government of India, under project number RTI400112, and in part by Google and SERB Research Grants.

[‡]University of Toronto, Canada. Email: `shubhangi.saraf@utoronto.ca`

Contents

1 Introduction

The main problem of interest in this work is that of polynomial factorization — *given a polynomial as input, output its decomposition into a product of irreducible polynomials.*

For this paper, we work in the setting where the input is a multivariate polynomial which is specified by a (small) algebraic circuit computing it, and we are over the field $\mathbb{Q}$ of rational numbers. This problem saw significant progress starting in the 1980s where a sequence of results culminating in the works of Kaltofen [Kal89] and Kaltofen & Trager [KT90] gave a randomized algorithm that when given a size s algebraic circuit C computing an n variate degree d polynomial over $\mathbb{Q}$, terminated in $\text{poly}(s, d, n)$ time and output algebraic circuits for all the irreducible factors of C (together with their multiplicities). A surprising fact that is implicit in these results is a *closure result* for polynomials computable by small circuits - all irreducible factors of C have algebraic circuits of size $\text{poly}(s, d, n)$. This is necessary for us to be able to entertain any hopes of having a polynomial time algorithm for this problem since a priori it is not even clear if there is a description of the output (namely, the irreducible factors of C) that is polynomially bounded in the input parameters s, d, n.

These polynomial factorization algorithms represent a significant landmark in our understanding of a fundamental problem in computational algebra on their own. However, in hindsight, the impact of this line of research seems to go far beyond this original context - these results and the techniques discovered in the course of their proofs have since found many diverse applications in algorithm design, pseudorandomness, coding theory and complexity theory, e.g. [Sud97, GS99, Ale05, Bog05, DGV24].

3

Among the most important problems in this broad area of polynomial factorization that continue to be open is that of derandomizing the results of Kaltofen [Kal89] and Kaltofen & Trager [KT90]. In fact, even before randomized factoring algorithms were studied for general circuits, they were studied in the setting of sparse polynomials. A beautiful work by von zur Gathen and Kaltofen [GK85] gave the first randomized factoring algorithm for sparse polynomials[1]. Even for sparse polynomials, the problem of derandomizing these factoring algorithms is of great interest and has received considerable attention over the last decade or two.

We now know from the work of Shpilka & Volkovich [SV10] that efficient deterministic polynomial factorization is at least as hard as efficient deterministic polynomial identity testing- the reduction simply being that to check whether a multivariate $f(\mathbf{x})$ is non-zero, we check if the polynomial $(f(\mathbf{x}) + yz)$ is irreducible. This connection clearly continues to hold for structured sub-classes for algebraic circuits like formulas, branching programs, sparse polynomials and constant-depth circuits. Thus, the task of derandomizing polynomial factorization for any such subclass of circuits must necessarily be preceded by a non-trivial deterministic polynomial identity testing algorithm for the subclass.

Perhaps a bit surprisingly, Kopparty, Saraf & Shpilka [KSS15] also showed a reduction in the other direction - efficient deterministic polynomial identity testing for algebraic circuits implies efficient deterministic polynomial factorization for algebraic circuits. This reduction between the problems of polynomial factorization and polynomial identity testing continue to hold in both the black box model (where we only have query access to the input algebraic circuits) and the white box model (where we can look inside the given circuits). An important aspect of the reduction in [KSS15] is that even for the task of factorizing polynomials computed by very structured circuits like formulas, constant-depth circuits or even sparse polynomials, the PIT instances that we encounter on the way are for seemingly much more powerful circuit classes like algebraic branching programs. Thus, non-trivial PIT algorithms for a structured circuit class do not immediately yield a non-trivial deterministic algorithm for factorization of polynomials computed in this class.

A very natural example of this phenomenon is that of sparse polynomials (which are depth 2 circuits) and even general constant-depth circuits. For the case of sparse polynomials, we have known polynomial-time PIT algorithms for a while from the work of Klivans & Spielman [KS01], and for arbitrary constant-depth circuits, we also now have non-trivial deterministic algorithms for polynomial identity testing of constant-depth circuits. These follow from the lower bounds for constant-depth circuits in the recent work of Limaye, Srinivasan and Tavenas [LST21] and the connections between hardness and derandomization for such circuits in the work of Chou, Kumar and Solomon [CKS19][2]. However, these deterministic polynomial identity tests do not seem to immediately imply non-trivial deterministic factorization results for either sparse polynomials

---

[1]The running time obtained was polynomial in the sparsity of the factors

[2]An alternative deterministic subexponential time algorithm for PIT for constant-depth circuits was given by Andrews and Forbes [AF22] in a subsequent work.

or polynomials computed by constant-depth circuits. In fact, the following seemingly simpler question also appears to be open.

Question 1.1. *Design a subexponential-time deterministic algorithm that when given a sparse polynomial as input, decides if it is irreducible.*

More generally, as Forbes & Shpilka mention (Questions 1.4 and 4.1 in [FS15]) in their survey on polynomial factorization, very natural questions around derandomization of polynomial factorization algorithms are wide open.

1.1 Our results

Our main result in this paper addresses this question and more generally, the question of deterministic polynomial factorization for constant-depth circuits. More precisely, we prove the following.

Theorem 1.2 (Informal version of Theorem 5.2). *For every constant $\varepsilon > 0$, constant $\Delta \in \mathbb{N}$ and sufficiently large n, the following statement is true.*

There is a deterministic algorithm that takes as input an algebraic circuit C over the field $\mathbb{Q}$ on n variables with depth Δ, and size and degree $\mathrm{poly}(n)$, runs in time $\exp(O(n^\varepsilon))$ and outputs algebraic circuits (of potentially unbounded depth) for all irreducible factors of C, together with their multiplicities.

This result in particular also gives the first subexponential deterministic factoring algorithm for sparse polynomials (which is the case when $\Delta = 2$).

At the heart of our proof is the deterministic construction of a variable reduction map that reduces the number of variables in a constant-depth circuit substantially, while preserving its *factorization profile*. The following theorem is our main technical result.[3]

Theorem 1.3 (Informal version of Theorem 5.1). *For every $\varepsilon > 0$, every constant $\Delta \in \mathbb{N}$ and all sufficiently large n, there is a polynomial map $\Gamma_{\varepsilon,\Delta} : \mathbb{Q}^{n^\varepsilon} \to \mathbb{Q}^n$ of degree $o(\log n)$ with the following properties.*

- *Given ε, Δ, the map $\Gamma_{\varepsilon,\Delta}$ can be constructed deterministically in time $\exp(O(n^\varepsilon))$.*

- *$\Gamma_{\varepsilon,\Delta}$ is a variable reduction map that preserves the irreducibility of n-variate polynomials that can be computed by depth Δ circuits of size and degree $\mathrm{poly}(n)$. In other words, an n-variate algebraic circuit C of size and degree $\mathrm{poly}(n)$ is irreducible if and only if the n^ε-variate polynomial $\hat{C}$ obtained from C by composing it with $\Gamma_{\varepsilon,\Delta}$ is irreducible.*

---

[3]The theorem follows immediately from Theorem 5.1 and our deterministic implementation of some (fairly standard) preprocessing steps in the factorization algorithm. In fact, by a slight modification of our proofs, a stronger version of this theorem can be shown to be true where the time complexity of constructing the map $\Gamma_{\varepsilon,\Delta}$ is quasipolynomially bounded in n. However, this does not improve the overall time complexity of the algorithm in Theorem 1.2 since some steps in the algorithm, for instance, that of deterministic factorization of n^ε variate polynomials of degree $\mathrm{poly}(n)$, still run in time $\exp(O(n^\varepsilon))$.

- *More generally, $\Gamma_{\varepsilon,\Delta}$ is a variable reduction map that preserves the irreducibility of the factors of n-variate polynomials that can be computed by depth Δ circuits of size and degree* $\mathrm{poly}(n)$.

Obtaining Theorem 1.2 from Theorem 1.3. The second item of the above theorem, when combined with known deterministic factorization algorithms for n^ε-variate polynomials of degree at most $\mathrm{poly}(n)$ that run in time $n^{O(n^\varepsilon)}$ (polynomial in the size of the dense representation of such polynomials), essentially gives us a subexponential time deterministic irreducibility testing algorithm for sparse polynomials, and more generally for constant-depth circuits. Similarly, the third item of Theorem 1.3 can be combined with known ideas in algorithms for polynomial factorization (with some effort) to give Theorem 1.2.

We end this section with a brief discussion of some quantitative and qualitative aspects of the main theorems.

Improvements in time complexity. Given the reduction from polynomial identity testing to polynomial factorization in [SV10], we know that one cannot hope to improve the running time of the algorithm in Theorem 1.2 significantly for general constant-depth circuits, unless we have significantly better deterministic polynomial identity testing algorithms for these problems. However, for the case of sparse polynomials, we have known polynomial time PIT algorithms for more than two decades [KS01], and thus we can, in principle, expect faster factorization or irreducibility testing algorithms for sparse polynomials. It is a very natural question to explore further.

Closure results for constant-depth circuits. We do not know if irreducible factors of polynomials with small constant-depth circuits (or even the more restricted class of sparse polynomials) must be computable by small constant-depth circuits, and Theorem 1.2 does not appear to shine any light on such a closure result. As a consequence, the outputs of the algorithm in Theorem 1.2 are general unbounded depth circuits of polynomial size. In fact, we do not even know of non-trivial upper bounds on the size of constant-depth circuits for irreducible factors of sparse polynomials. It would be very interesting to obtain such closure results or to gather some evidence that points towards such a statement being false.

Field dependence in Theorem 1.2. Our results in Theorem 1.2 are stated over the field of rational numbers. However, the results hold a little more generally - our proofs continue to work over any underlying field over which we have an efficient deterministic algorithm for factoring univariate polynomials, over which the lower bounds of Limaye, Srinivasan & Tavenas [LST21] for constant-depth circuits, the results of Andrews & Wigderson [AW24], and the Taylor-expansion-based techniques of Chou, Kumar & Solomon [CKS19] continue to work. The field of rational numbers satisfies all these properties. The results also continue to hold over finite fields of moderately large

characteristic (polynomially large in the degree and number of variables) since they satisfy all the properties stated above. However, for our presentation in this paper, we just work over the field of rational numbers and skip the minor technical changes needed to see the extension of the results to finite fields of moderately large characteristic.

1.2 Related prior work

Following the subexponential time deterministic PIT algorithms for constant-depth algebraic circuits that follow from the lower bounds in the work of Limaye, Srinivasan & Tavenas [LST21], there has been a renewed interest in obtaining deterministic algorithms for factorization of constant-depth circuits. This includes the results of Kumar, Ramanathan & Saptharishi [KRS23], who gave a subexponential time deterministic algorithm to compute all constant-degree factors of polynomials computed by constant-depth circuits and an alternative proof as well as a generalization of this result by Dutta, Sinhababu & Thierauf [DST24]. However, these results do not appear to give anything non-trivial for factors that are not low-degree.

Another result that is very relevant here is a work of Kumar, Ramanathan, Saptharishi & Volk [KRSV24] that gave a deterministic subexponential time algorithm that on input a constant-depth circuit outputs a list of circuits, of unbounded depth and with division gates, such that every irreducible factor of the input polynomial is computed by some circuit in this list. However, a significant drawback of this result is that the output list could contain circuits that do not correspond to any factor of the input, or aren't even valid circuits in the sense that they involve division by a circuit that computes an identically zero polynomial. In particular, their algorithm does not imply a deterministic subexponential time algorithm for testing irreducibility of a sparse polynomial since even when the input is irreducible, the algorithm could output a collection of circuits of tentative factors. However, in this case, none of the output circuits correspond to a true factor of the input polynomial. Unfortunately, since the depth of these output circuits is potentially large (and they contain division gates), we have no deterministic way of checking if there is a polynomial in the output list that corresponds to a true factor of the input. Thus, the question of pruning the output list deterministically in [KRSV24] to identify true factors is non-trivial and is essentially open.

A recent work of Andrews & Wigderson [AW24] gives efficient parallel algorithms (essentially constant-depth circuits) for a variety of problems related to polynomial factorization, such as GCD and LCM computation of polynomials. As a consequence of their techniques, they show that the resultant and the discriminant polynomials can be computed by constant-depth circuits of polynomial size, and the squarefree part of a polynomial with small constant-depth circuits is a small constant-depth circuit. The techniques of Andrews & Wigderson also give an alternative proof of the results in [KRSV24]. However, it suffers from the same drawback as [KRSV24] - some of the circuits in the output list might not correspond to any valid factors of the input, and might not even be valid circuits since they might have a division by an identically zero circuit built inside.

Even though the aforementioned results represent some interesting progress towards the question of obtaining deterministic factorization of constant-depth circuits continues, it seems fair to say that the original problem has largely remained open. In particular, these results do not say anything at all about even the question of sparse irreducibility testing (Question 1.1), which seems like a natural and possibly simpler intermediate step on the way to obtaining deterministic factorization of constant-depth circuits.

Factoring of sparse polynomials has been studied for a long time and it was initiated by the work of von zur Gathen and Kaltofen almost four decades ago [GK85]. This work gave the first randomized factoring algorithm for sparse polynomials, and the running time obtained was polynomial in the sparsity of the factors. Ever since this work, it has been an interesting question to obtain deterministic factoring algorithms for sparse polynomials, especially since we have known how to derandomize PIT for this class for a while.

There are special cases of sparse polynomials for which we do know some results on irreducibility testing and factoring. The work of Bhargava, Saraf & Volkovich [BSV18] gave a quasipolynomial time deterministic factoring algorithm for sparse polynomials where each variable has bounded degree. This extends prior works by Shpilka & Volkovich [SV10] which gave deterministic factoring algorithms for multilinear sparse polynomials and Volkovich [Vol17] which deterministic factoring algorithms for multiquadratic sparse polynomials.

As the main result of this paper (Theorem 1.2), we give a subexponential time deterministic algorithm for factorizing polynomials computable by small constant-depth circuits and, in particular, sparse polynomials (which are equivalent to depth-2 circuits). On the way to the proof, we obtain a deterministic subexponential time algorithm for testing the irreducibility of polynomials computable by small constant-depth circuits. In terms of techniques, we rely on some of the insights from prior work, e.g. [CKS19, KRS23, KRSV24, AW24] and introduce some new ideas on the way that might have further applications for problems of this nature. In particular, for our proofs, we crucially rely on technical observations about the structure of power series roots of polynomials with small constant-depth circuits, and their interaction with tools from hardness-vs-randomness for algebraic circuits, e.g. the Kabanets-Impagliazzo generator. As our main technical statement, we show that this generator, when invoked with a low-degree polynomial that is hard for constant-depth circuits (as constructed in [LST21]) not only preserves the non-zeroness of constant-depth circuits[4] (as was already known), but also preserves their irreducibility (and the irreducibility of their factors).

In the next section, we discuss these techniques in greater detail and give an overview of the proof of Theorem 1.2.

---

[4]For this, we need to assume that the circuit has undergone some initial preprocessing, which again is done using a similar generator.

2 Overview of proofs

Almost all factorization algorithms proceed with some initial pre-processing to guarantee the following requirements:

- $P(\mathbf{x}, z)$ is squarefree and monic in z.

- We have that $P(\mathbf{0}, 0) = 0$ and $\partial_z(P)(\mathbf{0}, 0) \neq 0$.

The above non-degeneracy conditions can typically be guaranteed relatively easily via a randomized algorithm. To do this deterministically, we use a recent result of Andrews & Wigderson [AW24] (see Theorem 4.4) to get a squarefree decomposition of P. In fact, their theorem says something more that is useful for us: the squarefree parts of P are computable by a small constant-depth circuit since P is computable by a small constant-depth circuit. Moreover, we can obtain this decomposition via a deterministic subexponential time algorithm that uses the now known deterministic PIT algorithms for constant-depth circuits [LST21, AF22].

We also translate the $\mathbf{x}$ variables appropriately (again using deterministic PIT for constant-depth circuits) to ensure that we are working with polynomials that are monic in z and each of the squarefree parts remains squarefree even when all the $\mathbf{x}$ variables are set to zero.

We then invoke a univariate factorization algorithm to find a root of $P(\mathbf{0}, z)$, which for this discussion we assume is zero. Furthermore, we replace each x_i with $x_i T$, for a fresh variable T, to work with a polynomial of the form $P(T, \mathbf{x}, z)$. Note that $P(0, \mathbf{x}, z) \in \mathbb{F}[z]$; we will refer to such polynomials as *T-regularized*.[5]

The general plan is to use Newton iteration, starting with $\varphi_0(T, \mathbf{x}) = 0$, to get "approximate roots" $\varphi_{k-1}(T, \mathbf{x})$ satisfying $\deg_T(\varphi_{k-1}) < k$ and $\varphi_{k-1}(T, \mathbf{x}) = \varphi_0(T, \mathbf{x}) \bmod T$ such that $P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x})) = 0 \bmod T^k$. We can obtain a small circuit for φ_{k-1}, although not necessarily one of constant depth. The hope is that, if k is large enough, then we can recover from φ_{k-1} a true factor of $P(T, \mathbf{x}, z)$.

2.1 Restricting to roots

Let us start by focusing on just extracting factors of $P(T, \mathbf{x}, z)$ of the form $(z - f(T, \mathbf{x}))$. If $f(T, \mathbf{x}) = \varphi_0(T, \mathbf{x}) \bmod T$, then uniqueness of Newton Iteration guarantees that $\varphi_{k-1}(T, \mathbf{x}) = f(T, \mathbf{x}) \bmod T^k$ as well. This therefore leads to the following natural algorithm:

1. Compute a circuit for $\varphi_{k-1}(T, \mathbf{x})$ for $k > \deg_T(P)$, with $\deg_T(\varphi_{k-1}) < k$, via Newton Iteration.

2. Check if $P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x})) = 0$.

---

[5]More generally, we say that a polynomial in $\mathbb{F}[T, \mathbf{x}, z]$ is *T-regularized* if every monomial with $\mathbf{x}$-degree at least one in the polynomial has T-degree at least one.

If it were the case that φ_{k-1} was actually computable by a constant-depth circuit, then the circuit $P(T, \mathbf{x}, \varphi_{k-1})$ is also constant-depth circuit and we can check for zeroness via known subexponential time PIT for constant-depth circuits. Unfortunately, we do not (yet) know if $\varphi_{k-1}(T, \mathbf{x})$ is computable by small constant-depth circuits.

The main insight is to notice that the circuit for φ_{k-1} that we obtain is structured enough that perhaps we can show that $P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x})) \overset{?}{=} 0$ can be tested via the same PIT nevertheless. We show that we can analyze the standard Kabanets-Impagliazzo generator instantiated with a suitable hard polynomial for such structured circuits and prove that it does preserve nonzeroness. The details can be found in Section 7.

2.2 Analyzing the KI generator on such circuits

The main structural lemma that drives our analysis is the following that shows that "low degree parts" of approximate roots can indeed be computed via not-too-large constant-depth circuits.

Lemma (Informal version of Lemma 6.3)**.** *Suppose $\varphi_{k-1}(T, \mathbf{x})$ with $\deg_T \varphi_{k-1} < k$ is an approximate z-root of $P(T, \mathbf{x}, z)$ as above. Then, for every integer $\ell \geq 0$, there is a circuit $C_\ell(T, \mathbf{x})$ of constant depth and "not-too-large" size that agrees with φ_{k-1} on all monomials of degree at most ℓ. That is,*

$$C_\ell(T, \mathbf{x}) = \varphi_{k-1}(T, \mathbf{x}) \bmod \langle \mathbf{x} \rangle^\ell,$$

where "not-too-large" is $\mathrm{poly}(\mathrm{size}(P), \deg(P)) \cdot (\log k)^{\mathrm{poly}(\ell)}$.

While this is reminiscent of a lemma of [CKS19] (Lemma 4.9) that allows one to argue that low-degree components of a root have small constant-depth circuits, there is an important difference in our statement: $\varphi_{k-1}(T, \mathbf{x})$ is a root modulo T^k, whereas we want a small constant-depth circuit to compute the root modulo $\langle \mathbf{x} \rangle^\ell$ i.e. a *different* set of variables. This difference demands a little more care (see Section 6.2.1), although morally we still use a Taylor-expansion based approach along the lines of [CKS19].

With the above lemma, we would be able to show that the Kabanets-Impagliazzo generator will indeed preserve the nonzeroness of $P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x}))$ when instantiated with a hard polynomial for constant-depth circuits.

Theorem (Informal version of Theorem 7.3)**.** *Let f be a polynomial that requires "large" constant-depth circuits. Then,*

$$P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x})) \neq 0 \implies P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x})) \circ \mathbf{KI}_f \neq 0$$

where $\mathbf{KI}_f : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{w}]$ is the Kabanets-Impagliazzo generator instantiated with the polynomial f.

We now briefly outline why this is true. Let us define $R(T, \mathbf{x}) := P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x}))$. To prove the above lemma, assume on the contrary that the generator $\mathbf{KI}_f$ did not preserve the non-zeroness of $R(T, \mathbf{x})$. By the standard hybrid argument (with additional substitutions), there are "easy-to-compute" polynomials f_i's such that

$$R(f_1, f_2, \ldots, f_{i-1}, x_i, f_{i+1}, \ldots, f_n, T) \neq 0$$
$$\text{but } R(f_1, f_2, \ldots, f_{i-1}, f, f_{i+1}, \ldots, f_n, T) = 0.$$

$(f_{i+1}, \ldots, f_n$ are just field constants). Therefore, the polynomial $x_i - f$ divides the polynomial $R(f_1, \ldots, f_{i-1}, x_i, f_{i+1}, \ldots, f_n, T)$. At this point, we make use of the following result by Chou, Kumar and Solomon (stated informally).

Theorem (Chou-Kumar-Solomon (follows immediately from Lemma 4.9)). *Let $R(\mathbf{w}, y)$ be a non-zero polynomial with $R(\mathbf{w}, g(\mathbf{w})) = 0$ for some polynomial g of degree at most ℓ, with $\partial_y R(0, g(0)) \neq 0$. For all $i \leq \ell$, suppose there are small circuits $C_i(\mathbf{w}, y)$ such that*

$$C_i(\mathbf{w}, y) = \partial_{y^i} R \bmod \langle \mathbf{w} \rangle^\ell.$$

If ℓ is "small", then g is a "small" composition of the circuits $C_1, \ldots, C_\ell$. In particular, if ℓ is small and each C_i is a small, constant-depth circuit, then g is also computable by a "small-ish", constant-depth circuit.

For our setting, let $\tilde{R}(\mathbf{w}, y, T) = R(f_1, f_2, \ldots, f_{i-1}, y, f_{i+1}, \ldots, f_n, T)$ which has $y - f(\mathbf{w})$ as a factor. Note that this is not a constant-depth circuit as $R(T, \mathbf{x}) = P(T, \mathbf{x}, \varphi_{k-1}(T, \mathbf{x}))$ and φ_{k-1} is not known to have a constant-depth circuits. However, if $\deg(f)$ is small enough, then the above theorem of Chou, Kumar and Solomon asserts that we can construct a constant-depth circuit for $(y - f(\mathbf{w}))$ if we are able to build constant-depth circuits for $\mathrm{Hom}_{\leq \deg(f)} \left(\partial_{y^i} \tilde{R}(\mathbf{w}, y, T) \right)$. Fortunately, our main structural lemma states that low-degree homogeneous parts of φ_{k-1} do have not-too-large constant-depth circuits. Putting this together, we are able to assert that $f(\mathbf{w})$ must also be computable by a not-too-large constant-depth circuit, thus contradicting the hardness of f.

2.3   Computing general factors

For computing general factors, at a high level, the proof proceeds in three broad steps. We first show a way of characterizing irreducibility of polynomials by (exponentially many) divisibility tests involving the power series roots of these polynomials. We then show that these divisibility tests can be reduced to PIT instances for circuits that might not be constant-depth, but have some additional structure. And finally, in spite of being unable to show that these PIT instances are for constant-depth circuits, we manage to show that the Kabanets-Impagliazzo hitting set generator [KI04] invoked with low degree hardness for constant-depth circuits from [LST21] preserves the

non-zeroness of these PIT instances. This part of the proof again builds upon the techniques from [CKS19]. We now discuss these steps in a bit more detail and refer to Section 8 for the full proof.

The main technical insight of our proof is that the Kabanets-Impagliazzo generator invoked with low-degree hard polynomials for constant-depth circuits in fact preserves irreducibility of small constant-depth circuits and their factors. Let us consider the polynomial $P(\mathbf{0}, 0, z)$ over the algebraic closure $\overline{\mathbb{Q}}$, and let us assume this splits as $(z - \zeta_1) \cdots (z - \zeta_d)$. Let $\varphi^{(1)}(T, \mathbf{x}), \ldots, \varphi^{(d)}(T, \mathbf{x})$ be the approximate root lifted from ζ_i — i.e., it satisfies $P(T, \mathbf{x}, \varphi^{(i)}) = 0 \bmod T^k$ and $\varphi^{(i)}(\mathbf{0}, 0) = \zeta_i$, where k is large enough. Then, each true factor $Q(T, \mathbf{x}, z)$ of $P(T, \mathbf{x}, z)$ corresponds to some subset S_Q of these approximate roots. That is, we must have

$$Q(T, \mathbf{x}, z) = \prod_{i \in S_Q} (z - \varphi^{(i)}(T, \mathbf{x})) \text{ trunc } T^k$$

where " trunc T^k" denotes the operation of discarding all monomials that have degree k or more in T. As a consequence, if $P(T, \mathbf{x}, z)$ is indeed irreducible, then for every subset $\varnothing \neq S \subsetneq [d]$ we have $Q_S(T, \mathbf{x}, z)$ does not divide $P(T, \mathbf{x}, z)$ as a polynomial, where

$$Q_S(T, \mathbf{x}, z) = \prod_{i \in S} (z - \varphi^{(i)}(T, \mathbf{x})) \text{ trunc } T^k .$$

Turns out, the non-divisibility of such Q_S, P can be expressed as an appropriate PIT via standard reductions from divisibility testing to PIT [For15, AW24]. Once again, this is not a PIT of a constant-depth circuit due to the presence of the $\varphi^{(i)}(T, \mathbf{x})$ sub-circuits which are not known to have small constant-depth circuits. Nevertheless, since we are able to show that $\varphi^{(i)}(T, \mathbf{x})$ has enough structure to allow us to build small constant-depth circuits for their "low-degree" components, we are able to argue that the Kabanets-Impagliazzo generator, when instantiated with a sufficiently hard low-degree polynomial, will preserve the non-zeroness of such PIT instances. Furthermore, our algorithms do not proceed by directly derandomizing these PIT instances since there are exponentially many of them to deal with. We only use this reduction in the analysis of our algorithms.

Overall, this yields a variable reduction (to n^ε variables, for any $\varepsilon > 0$) that allows us to preserve the factorization pattern of any constant-depth circuit. From this, recovering circuits (although not necessarily constant-depth) for the factors is relatively straightforward, given the known algorithms for polynomial factorization.

Organization

The rest of the paper is organized as follows. We begin by recalling some standard notations and preliminaries in Section 3, and some preliminaries related to polynomial factorization in Section 4. We describe the details of our algorithms in Section 5, and discuss their analysis, assuming some

technical results in Section 9. We discuss the proof of these technical results in detail in Section 6, Section 7 and Section 8.

For readers familiar with the general area of algebraic circuit complexity and some familiarity with polynomial factorization algorithms, we recommend skipping the preliminaries and going to Section 5, and then referring to Section 3 and Section 4 as and when needed.

3 Standard preliminaries

3.1 Notation

- Throughout the paper $\mathbb{F}$ denotes a field and $\mathbb{Q}$ denotes the field of rational numbers.

- We use letters like x, y, z to refer to formal variables and letters like a, b, c as constants, as would be clear from the context. Boldface letters $\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{a}, \mathbf{b}$ etc. refer to tuples of such objects. The arity of the tuple is generally specified unless it is clear from the context.

- We say that a multivariate polynomial P is *monic* in a specific variable y if the coefficient of the highest degree monomial in y in P is a non-zero field constant.

- An algebraic circuit over a field $\mathbb{F}$ on variables $\mathbf{x}$ is a directed acyclic graph with internal nodes being labeled by product ($\times$) or sum ($+$), and the leaves (nodes of in-degree zero) being labeled by variables in $\mathbf{x}$ or constants from $\mathbb{F}$. All the fan-ins are unbounded.

  Such a circuit computes a polynomial in a natural sense - a leaf computes the polynomial that is equal to its label, a sum gate computes the sums of its inputs and a product gate computes the product of its inputs. The size of an algebraic circuit equals the number of edges in it and the depth equals the length of the longest path from a leaf to an output node (a node of out-degree zero). We refer to the surveys [SY10, Sap15] for detailed discussions on algebraic circuits.

- Polynomial identity testing or PIT refers to the decision problem where the input is an algebraic circuit, and the goal is to decide if the polynomial computed by the circuit is identically zero.

- For a polynomial P and a variable y, $\deg_y(P)$ refers to the degree of P with respect to y. Similarly, for a tuple $\mathbf{x}$ of variables, $\deg_{\mathbf{x}}(P)$ refers to the total degree of P with respect to variables in $\mathbf{x}$.

3.2 Truncation

We start with the important, although non-standard definitions that are helpful in succinctly expressing our technical statements in the paper.

Definition 3.1 (Polynomial truncation)**.** *For a polynomial $Q(\mathbf{x}) \in R[\mathbf{x}]$ and a positive integer k, Q trunc $\langle \mathbf{x} \rangle^k$ denotes the unique polynomial $\tilde{Q}$ with $\mathbf{x}$-degree less than k such that $\tilde{Q} \equiv Q \bmod \langle \mathbf{x} \rangle^k$. We extend this notation to $Q(T, \mathbf{x}) \in \mathbb{F}[\mathbf{x}][T]$ (as well as power-series in $\mathbb{F}[\mathbf{x}][\![T]\!]$) and use Q trunc T^k to denote the truncation interpreting $Q(T, \mathbf{x}) \in R[T]$ (or $R[\![T]\!]$) for $R = \mathbb{F}[\mathbf{x}]$.* ◇

While it is common to slightly overload notation and use mod $\langle \mathbf{x} \rangle^k$ to also denote trunc $\langle \mathbf{x} \rangle^k$, we choose to separate these notations for reasons of clarity.

3.3 Polynomial Identity Lemma

We now recall the statement of the polynomial identity lemma.

Lemma 3.2 (Polynomial Identity Lemma [Ore22, DL78, Sch80, Zip79])**.** *Let P be an n-variate non-zero polynomial of degree at most d over a field $\mathbb{F}$. And, let S be any subset of $\mathbb{F}$.*

Then, the number of zeroes of P on the product set $S \times S \times \cdots \times S$ is at most $d|S|^{n-1}$. In particular, if $|S| > d$, then P is non-zero on at least one point on $S \times S \times \cdots \times S$.

3.4 Interpolation

We now recall the standard interpolation lemma for extracting coefficients of univariates from their evaluations.

Lemma 3.3 (Interpolation (cf. [Sap15, Lemma 5.3]))**.** *Let R be a commutative ring that contains a field $\mathbb{F}$ of at least $d + 1$ elements, and let $\alpha_0, \ldots, \alpha_d$ be distinct field elements in $\mathbb{F}$. Then, for each $i \in \{0, \ldots, d\}$, there are constants $\beta_{i0}, \ldots, \beta_{id} \in \mathbb{F}$ such that for every $f(x) = f_0 + f_1 x + \cdots + f_d x^d \in R[x]$ we have*

$$f_i = \sum_{j=0}^{d} \beta_{ij} \cdot f(\alpha_j).$$

The following corollary invokes this lemma in some of the contexts that appear in our proof. The proof is immediate from the lemma.

Corollary 3.4 (Standard consequences of interpolation)**.** *Let $\alpha_0, \ldots, \alpha_d$ be distinct elements in $\mathbb{F}$. Then,*

1. ***[Partial derivatives]*** *If $C(\mathbf{x}, y)$ has degree d in the variable y, then the i-th order partial derivative of C with respect to y can be expressed as an $\mathbb{F}[y]$-linear combination of $\{C(\mathbf{x}, \alpha_j) : j \in \{0, \ldots, d\}\}$. That is, there are polynomials $\mu_0(y), \ldots, \mu_d(y)$ (not depending on C) of degree at most d such that*

$$\partial_{y^i} C(\mathbf{x}, y) = \mu_0(y) \cdot C(\mathbf{x}, \alpha_0) + \cdots + \mu_d(\mathbf{y}) \cdot C(\mathbf{x}, \alpha_d).$$

2. ***[Homogeneous components]*** *Let $C(\mathbf{x})$ be a degree d polynomial. Then, for any subset $\mathbf{x}_S \subseteq \mathbf{x}$ and any $i \in [d]$, the degree i homogeneous part of C with respect to $\mathbf{x}_S$, denoted by $\mathrm{Hom}_{\mathbf{x}_S, i}(C)$, can be*

expressed as

$$\text{Hom}_{\mathbf{x}_S,i}(C) = \sum_{j=0}^{d} \beta_{i,j} \cdot C(\alpha_j \cdot \mathbf{x}_S, \mathbf{x}_{\overline{S}})$$

for some constants $\beta_{i,j} \in \mathbb{F}$ (not depending on C).

3. *[**Truncation**] Let $C(\mathbf{x})$ be a degree d polynomial and let $\mathbf{x}_S \subseteq \mathbf{x}$ be a subset of variables. Then, for any $\ell \in [d]$, the truncation $C(\mathbf{x})$ trunc $\langle \mathbf{x}_S \rangle^{\ell+1}$ can be expressed as*

$$C(\mathbf{x}) \text{ trunc } \langle \mathbf{x}_S \rangle^{\ell+1} = \sum_{j=0}^{d} \gamma_{\ell,j} \cdot C(\alpha_j \cdot \mathbf{x}_S, \mathbf{x}_{\overline{S}})$$

for some constants $\gamma_{\ell,j} \in \mathbb{F}$ (not depending on C).

In particular, if C is computable by a size s, depth Δ circuit, then all of the above operations yield a circuit of size $\text{poly}(d,s)$ and depth $\Delta + O(1)$.

3.5 Strassen's division elimination

We recall a classical theorem of Strassen for division elimination in algebraic circuits.

Theorem 3.5 (Division elimination in algebraic circuits [Str73])**.** *Let C be an algebraic circuit of size s with division gates that computes a multivariate polynomial P of degree d over any sufficiently large field $\mathbb{F}$.*

Then, there is an algebraic circuit $\hat{C}$ of size at most $\text{poly}(s,d)$ that computes the polynomial P and does not have any division gates.

As a matter of notation, algebraic circuits throughout this paper do not have division gates. We explicitly mention if we have to deal with circuits with division gates at any point in our proof.

In our proofs, we rely on the following consequence of the above theorem. The proof easily follows from the standard proof of Theorem 3.5, for instance in [SY10].

Lemma 3.6 (Algorithmic division elimination)**.** *Let $\mathbb{F}$ be any field.*

There is a deterministic algorithm that takes as input algebraic circuits of size at most s computing n variate polynomials A, B over $\mathbb{F}$, a point $\mathbf{u} \in \mathbb{F}^n$ and a degree parameter d such that (a) B divides A, (b) the quotient A/B has degree at most d, and (c) $B(\mathbf{u}) \neq 0$, and outputs a (division free) algebraic circuit for the quotient A/B. Moreover, the algorithm runs in time $\text{poly}(s,d)$.

3.6 Resultant and Discriminant

The notion of resultant and its close connection to GCD of two univariate polynomials plays an important role in our proof (and in most polynomial factorization algorithms). We start by recalling the definition.

15

Definition 3.7 (Sylvester Matrix and Resultant). *Let $\mathbb{F}$ be any field, and let P and Q be univariates over $\mathbb{F}$ of degree equal to $a \geq 1$ and $b \geq 1$ respectively.*

Let $\Gamma_{P,Q} : \mathbb{F}^b \times \mathbb{F}^a \to \mathbb{F}^{a+b}$ be the $\mathbb{F}$ linear map that maps a pair (U, V) of univariates over $\mathbb{F}$ with degree of U at most $(b-1)$ and degree of V at most $(a-1)$ to the polynomial $(UP + VQ)$ of degree at most $(a+b-1)$.

Then, the Sylvester matrix of P and Q is the $(a+b) \times (a+b)$ matrix for the $\mathbb{F}$-linear map $\Gamma_{P,Q}$, when the inputs and the outputs are represented as their coefficient vectors.

And, the Resultant of P, Q is defined as the determinant of the Sylvester Matrix of P and Q. ◊

Clearly, the entries of the Sylvester matrix as defined above are the coefficients of P and Q.

In some applications in this paper, we end up invoking the notion of the resultant while working with multivariates P, Q. In these applications, we think of these multivariates as univariates in one of the variables, with the coefficients coming from the field of rational functions in the other variables. Unless otherwise clear from the context, we indicate the variable with respect to which these definitions are invoked.

The resultant has a deep and extremely useful connection to the GCD of two polynomials as the following classical theorem indicates. We refer to Chapter 6 in the book [vzGG13] for a proof.

Theorem 3.8 (Resultants and GCD [vzGG13, Corollary 6.20]). *Let $\mathcal{R}$ be a unique factorization domain, and let $P, Q \in \mathcal{R}[z]$ be non-zero polynomials. Then, $\deg_z(\gcd(P,Q)) \geq 1$ if and only if $\mathrm{Res}_y(P, Q) = 0$, where $\gcd(P,Q) \in \mathcal{R}[z]$ and $\mathrm{Res}_z(P, Q) \in \mathcal{R}$. Moreover, there exist polynomials $A, B \in \mathcal{R}[z]$ satisfying $\mathrm{Res}_z(P, Q) = AP + BQ$.*

A special case of the resultant that is very natural to study is when we consider the resultant of a polynomial $P(z)$ and its derivative $\frac{dP}{dz}$. This resultant is referred to as the *discriminant* of P, and unless the derivative vanishes for trivial reasons (for instance over fields of small characteristic), the discriminant captures the squarefreeness of P. More precisely, we have the following theorem.

Theorem 3.9 (Discriminant and squarefreeness [DSS22, Lemma 12]). *Let $\mathbb{F}$ be any field of characteristic zero, and let $P(z)$ be a univariate over $\mathbb{F}$ of degree at least one. Then, $P(z)$ is squarefree if and only if its discriminant $\mathrm{Res}_z(P, \frac{\partial P}{\partial z})$ is non-zero.*

A beautiful result of Andrews and Wigderson shows that the resultant can be computed by a constant-depth circuit.

Theorem 3.10 (Computing resultants via constant-depth circuits [AW24, Theorem 6.1]). *Let $\mathbb{F}$ be a field of characteristic zero. For a fixed $\Delta \in \mathbb{N}$, there is a family of depth-Δ circuits $\{C_n\}_{n \in \mathbb{N}}$ with size $\leq \mathrm{poly}(n)$ such that the following is true. If $f, g \in \mathbb{F}[z]$ are degree-n univariate polynomials given by their coefficients, then C_n takes the coefficients of f and g as input, and computes the resultant $\mathrm{Res}_z(f, g)$.*

3.7 Gauss' lemma

The following basic lemma of Gauss is useful for us in our proof.

Lemma 3.11 (Gauss' lemma [vzGG13, Section 6.2, Corollary 6.10])**.** *Let $\mathcal{R}$ be a unique factorization domain with the field of fractions $\mathcal{K}$. Then, a monic polynomial $P(z)$ is irreducible in $\mathcal{R}[z]$ if and only if it is irreducible in $\mathcal{K}[z]$.*
In particular, for any monic $P(z)$, the factorization of $P(z)$ into its irreducible factors in $\mathcal{R}[z]$ is identical to the factorization of $P(z)$ into its irreducible factors in $\mathcal{K}[z]$.

Moreover, for a monic $P(z) \in \mathcal{K}[z]$, its factors can also be assumed to be monic in z without loss of generality.

3.8 The Kabanets-Impagliazzo generator

The Kabanets-Impagliazzo hitting set generator [KI04] for algebraic circuits is an adaptation of the Nisan-Wigderson generator in classical complexity to the algebraic setting. It allows us to obtain non-trivial derandomization for PIT for algebraic circuits from sufficiently hard explicit polynomial families. More precisely, Kabanets & Impagliazzo showed in [KI04] that given an explicit polynomial family that requires exponential size algebraic circuits, there is a deterministic algorithm for PIT for algebraic circuits (with size and degree polynomially bounded in the number of variables) that runs in quasipolynomial time.

This generator and the details of its analysis play an important role in the proofs in this paper. We start by recalling the notion of combinatorial designs, and then define the generator.

Definition 3.12 (Combinatorial designs)**.** *Let $n, \sigma, \mu, \rho \in \mathbb{N}$. A family of subsets $\mathcal{S} = (S_1, \ldots, S_n)$ is a (n, σ, μ, ρ) design if:*

- *$\forall i \in [n] : S_i \subseteq [\mu]$*

- *$\forall i \in [n] : |S_i| = \sigma$*

- *For any $i, j \in [n]$ s.t. $i \neq j : |S_i \cap S_j| < \rho$*

$\Diamond$

The following theorem gives an explicit construction of such designs.

Lemma 3.13 (Explicit construction of designs [NW94])**.** *For any positive integers n, σ with $n < 2^{\sigma}$, there exists an explicit (n, σ, μ, ρ)-design with $\mu = \dfrac{\sigma^2}{\log n}$ and $\rho = \log n$.*
Moreover, each subset inside the design can be computed in $\mathrm{poly}(n, 2^{\mu})$ time deterministically.

Definition 3.14 (KI-generator $\mathbf{KI}_{g,\mathcal{S}}$ [KI04])**.** *Let $\mathbb{F}$ be a field, n, σ be positive integers with $n < 2^{\sigma}$ and $g(\mathbf{x})$ be a σ-variate polynomial. Let $\mathcal{S} = (S_1, \ldots, S_n)$ be an explicit (n, σ, μ, ρ)-design from Lemma 3.13.*

The Kabanets-Impagliazzo generator given by g and $\mathcal{S}$ is a polynomial map $\mathbf{KI}_{g,\mathcal{S}} : \mathbb{F}^{\mu} \to \mathbb{F}^n$ defined by $\mathbf{w} \mapsto (g_1(\mathbf{w}), \ldots, g_n(\mathbf{w}))$, where for each $i \in [n]$, $g_i(\mathbf{w}) := g(\mathbf{w}_{S_i})$ for $\mathbf{w}_{S_i} := \{w_j : j \in S_i\}$. Thus, $\mathbf{KI}_{g,\mathcal{S}}$ also defines a homomorphism from $\mathbb{F}[x_1, \ldots, x_n]$ to $\mathbb{F}[w_1, \ldots, w_\mu]$ that maps each x_i to $g_i(\mathbf{w})$. $\diamondsuit$

3.9 Lower bounds and PIT for constant-depth circuits

We now recall the results of [LST21] that prove superpolynomial lower bounds for constant-depth circuits.

Theorem 3.15 (Lower bounds for constant-depth circuits [LST21, Corollary 4])**.** *Suppose $n, d \in \mathbb{N}$ with $d \leq \log n / 100$ and $\mathbb{F}$ is a field with $\mathrm{char}(\mathbb{F}) = 0$ or greater than d. Then, for any product-depth $\Delta \in \mathbb{N}$, there exists an explicit n-variate polynomial $P(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d such that any algebraic circuit of product-depth at most Δ must have size at least $n^{d^{\exp(-O(\Delta))}}$.*

In [CKS19], it was shown that explicit *low degree* hard polynomials for constant-depth circuits as shown in the above theorem imply non-trivial deterministic PIT algorithms for constant-depth algebraic circuits. An alternative route to achieving the same result was shown later by Andrews & Forbes. We recall this theorem below.

Theorem 3.16 (Subexponential time deterministic PIT for constant-depth circuits [LST21, AF22])**.** *Let $\varepsilon > 0$ be a real number and $\mathbb{F}$ a field of characteristic 0. Let C be an algebraic circuit of size $s \leq \mathrm{poly}(n)$ and depth $\Delta = o(\log \log \log n)$, computing an n-variate polynomial . Then, there is a deterministic algorithm that can decide whether the polynomial computed by C is identically zero or not, in time $(s^{\Delta+1} \cdot n)^{O(n^\varepsilon)}$.*

4 Preliminaries for polynomial factorization

4.1 Regularized polynomials

Definition 4.1 (*T*-regularized polynomials and non-degenerate, truncated, approximate *z*-roots of order *k*)**.** *Let $P(T, \mathbf{x}, z)$ be a polynomial in $\mathbb{F}[T, \mathbf{x}, z]$ that is monic in the variable z, and $\Phi(T, \mathbf{x}) \in \mathbb{F}[\mathbf{x}][\![T]\!]$ be a power series.*

- $\Phi(T, \mathbf{x})$ *is said to be an* approximate *z*-root of order *k* with respect to *T* if

  $$P(T, \mathbf{x}, \Phi(T, \mathbf{x})) \equiv 0 \bmod T^k.$$

  Moreover, an approximate z-root Φ of order k is truncated *if $\deg_T \Phi < k$. Throughout the paper, approximate roots will be defined modulo T^k. For the sake of brevity, we sometimes refer to an approximate z-root of order k with respect to T as just an approximate z-root of order k.*

18

- $P(T, \mathbf{x}, z)$ *is T-regularized with respect to z if $P(0, \mathbf{x}, z) \in \mathbb{F}[z]$, that is, every monomial that depends on $\mathbf{x}$ is divisible by T.*

- $\Phi(T, \mathbf{x})$ *is a* non-degenerate *approximate z-root of $P(T, \mathbf{x}, z)$ if*

$$(\partial_z P)(0, \mathbf{0}, \alpha) \neq 0$$

  where $\alpha = \Phi(0, \mathbf{0}) \in \mathbb{F}$; that is, the constant term of Φ is not a repeated root of $P(0, \mathbf{0}, z)$. ◇

The following simple observation follows immediately from the above definitions.

Observation 4.2. *Let $P(T, \mathbf{x}, z)$ be a polynomial in $\mathbb{F}[T, \mathbf{x}, z]$ that is monic in the variable z, T-regularized, and let $\Phi(T, \mathbf{x}) \in \mathbb{F}[\mathbf{x}][\![T]\!]$ be a power series.*
If $\Phi(T, \mathbf{x})$ is an approximate z-root of $P(T, \mathbf{x}, z)$, then $\Phi(0, \mathbf{x})$ – a root of the univariate $P(0, \mathbf{x}, z) \in \mathbb{F}[z]$ – is a scalar in $\mathbb{F}$, implying that $\Phi(0, \mathbf{x}) = \Phi(0, \mathbf{0})$. Moreover, $(\partial_z P)(0, \mathbf{0}, z) = (\partial_z P)(0, \mathbf{x}, z)$.

4.2 Squarefree decomposition

We say that a polynomial P is *squarefree* if it is not divisible by the square of another polynomial. In particular, every irreducible factor of P appears with multiplicity one in the unique factorization of P.

We now define the notion of squarefree decomposition of a polynomial.

Definition 4.3 (Squarefree decomposition). *Let $F \in \mathbb{F}[\mathbf{x}]$ be a polynomial such that $F(\mathbf{x}) = \prod_{i=1}^{m} G_i(\mathbf{x})^{e_i}$. Let $r = \max_{i \in [m]} e_i$, where each G_i is irreducible. Then the squarefree decomposition of F is $(F_1, F_2, \ldots, F_r)$, where for each $i \in [r]$, $F_i := \prod_{j \in [m]: e_j = i} G_j$.* ◇

The following theorem of Andrews & Wigderson shows that squarefree parts of a polynomial computable by a small constant-depth circuit have small constant-depth circuits, and moreover, we can compute such a decomposition deterministically given an appropriate PIT oracle.

Theorem 4.4 (Squarefree decomposition [AW24]). *Let $\mathbb{F}$ be a field of characteristic zero or characteristic greater than D. Let $\mathcal{O}$ be an oracle that solves polynomial identity testing for constant-depth circuits. Then, there is a deterministic polynomial-time algorithm with oracle access to $\mathcal{O}$ which does the following:*

1. *The algorithm receives as a input a constant-depth circuit that computes a polynomial F of degree D.*

2. *The algorithm outputs a collection of constant-depth circuits $C_1, \ldots, C_r$ such that C_i computes F_i, where $(F_1, \ldots, F_r)$ is the squarefree decomposition of F.*

4.3 Newton iteration

We now recall various flavors of Newton iteration that we use in our proofs.

Lemma 4.5 (Newton iteration with linear convergence [CKS19, Lemma 5.1]). *Let $R = \mathbb{F}[\mathbf{x}]$ be a polynomial ring, and let $H(\mathbf{x}, z) \in R[z]$. Suppose $\varphi \in \mathbb{F}[\![\mathbf{x}]\!]$ is a power-series such that $H(\mathbf{x}, \varphi) = 0 \mod \langle \mathbf{x} \rangle^m$ and $\partial_z H(\mathbf{0}, \varphi(\mathbf{0})) \neq 0$. Then,*

$$\varphi' := \varphi - \frac{H(\mathbf{x}, \varphi)}{\partial_z H(\mathbf{0}, \varphi(\mathbf{0}))}$$

satisfies $H(\mathbf{x}, \varphi') = 0 \mod \langle \mathbf{x} \rangle^{m+1}$ and $\varphi' = \varphi \mod \langle \mathbf{x} \rangle^m$. Furthermore, such an extension φ' of φ is unique in the sense that any φ'' that satisfies $H(\mathbf{x}, \varphi'') = 0 \mod \langle \mathbf{x} \rangle^{m+1}$ and $\varphi'' = \varphi \mod \langle \mathbf{x} \rangle^m$ must satisfy

$$\varphi' = \varphi'' \mod \langle \mathbf{x} \rangle^{m+1}.$$

Corollary 4.6 ([CKS19, Corollary 5.5], [KRSV24, Lemma 3.1]). *Let $R = \mathbb{F}[\mathbf{x}]$ be a polynomial ring, and let $H(\mathbf{x}, z) \in R[z]$ be a polynomial of degree D and a circuit of size s. Suppose $u \in \mathbb{F}$ such that:*

$$H(\mathbf{0}, u) = 0$$
$$\frac{\partial H}{\partial z}(\mathbf{0}, u) \neq 0$$

Then for every $k \in \mathbb{N}$, there is a unique truncated, non-degenerate, approximate z-root $\Phi_k(\mathbf{x})$ of order k with respect to $\mathbf{x}$ for $H(\mathbf{x}, z)$, satisfying $\Phi_k(\mathbf{0}) = u$. Moreover, there is a deterministic algorithm that runs in time $\text{poly}(s, D, k)$ and outputs a circuit of size $\text{poly}(s, D, k)$ for $\Phi_k(\mathbf{x})$.

Lemma 4.7 (Newton iteration with quadratic convergence [vzGG13, Lemma 9.21, Lemma 9.27]). *Let $R = \mathbb{F}[\mathbf{x}]$ be a polynomial ring, and let $H(\mathbf{x}, z) \in R[z]$.*

Suppose $\varphi \in \mathbb{F}[\![\mathbf{x}]\!]$ is a power-series such that $H(\mathbf{x}, \varphi) = 0 \mod \langle \mathbf{x} \rangle^m$ and $\partial_z H(\mathbf{0}, \varphi(\mathbf{0})) \neq 0$. Then, for any power series $\sigma \in \mathbb{F}[\![\mathbf{x}]\!]$ satisfying

$$\sigma(\mathbf{x}) = \frac{1}{\partial_z H(\mathbf{x}, \varphi)} \mod \langle \mathbf{x} \rangle^m$$

we have that $\varphi' := \varphi - H(\mathbf{x}, \varphi)\sigma$ satisfies $H(\mathbf{x}, \varphi') = 0 \mod \langle \mathbf{x} \rangle^{2m}$ and $\varphi' = \varphi \mod \langle \mathbf{x} \rangle^m$. Furthermore, such an extension φ' of φ is unique in the sense that any φ'' that satisfies $H(\mathbf{x}, \varphi'') = 0 \mod \langle \mathbf{x} \rangle^{2m}$ and $\varphi'' = \varphi \mod \langle \mathbf{x} \rangle^m$ must satisfy

$$\varphi' = \varphi'' \mod \langle \mathbf{x} \rangle^{2m}.$$

Lemma 4.8 (Quadratic-convergence Newton Iteration without divisions). *Let $R = \mathbb{F}[\mathbf{x}]$ be a polynomial ring, and let $H(\mathbf{x}, z) \in R[z]$. Suppose there exists $\alpha \in \mathbb{F}$ such that $H(\mathbf{0}, \alpha) = 0$ and*

$\partial_z H(\mathbf{0}, \alpha) = \beta \neq 0$. For each $i \geq 0$, define polynomials $\varphi_i, \sigma_i \in \mathbb{F}[\mathbf{x}]$ as follows:

$$\varphi_0 := \alpha, \qquad\qquad\qquad \sigma_0 := (1/\beta),$$
$$\text{For } i \geq 0, \quad \varphi_{i+1} := \varphi_i - H(\mathbf{x}, \varphi_i) \cdot \sigma_i, \qquad \sigma_{i+1} := 2\sigma_i - \sigma_i^2 \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}).$$

Then $H(\mathbf{x}, \varphi_i) = 0 \bmod \langle \mathbf{x} \rangle^{2^i}$, $\varphi_{i+1} = \varphi_i \bmod \langle \mathbf{x} \rangle^{2^i}$, and $\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_i) = 1 \bmod \langle \mathbf{x} \rangle^{2^i}$.

Proof. We prove this by induction on i. The base case $i = 0$ follows by definition.
Now suppose that for some $i \geq 0$, $H(\mathbf{x}, \varphi_i) = 0 \bmod \langle \mathbf{x} \rangle^{2^i}$ and $\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_i) = 1 \bmod \langle \mathbf{x} \rangle^{2^i}$.
By Lemma 4.7, it follows that $\varphi_{i+1} := \varphi_i - H(\mathbf{x}, \varphi_i) \cdot \sigma_i$ satisfies $H(\mathbf{x}, \varphi_{i+1}) = 0 \bmod \langle \mathbf{x} \rangle^{2^{i+1}}$ and $\varphi_{i+1} = \varphi_i \bmod \langle \mathbf{x} \rangle^{2^i}$.

$$
\begin{aligned}
\sigma_{i+1} \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}) - 1 &= (2\sigma_i - \sigma_i^2 \cdot \partial_z H(\mathbf{x}, \varphi_{i+1})) \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}) - 1 \\
&= (\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}) - 1) - ((\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}))^2 - \sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_{i+1})) \\
&= -(\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_{i+1}) - 1)^2 \\
&= 0 \bmod \langle \mathbf{x} \rangle^{2^{i+1}}
\end{aligned}
$$

where the last equality follows because $\varphi_{i+1} = \varphi_i \bmod \langle \mathbf{x} \rangle^{2^i}$ and $\sigma_i \cdot \partial_z H(\mathbf{x}, \varphi_i) - 1 = 0 \bmod \langle \mathbf{x} \rangle^{2^i}$. $\qquad\square$

4.4 A lemma of Chou, Kumar & Solomon

The following technical lemma of Chou, Kumar & Solomon [CKS19] is used in the analysis of Kabanets-Impagliazzo generator for constant-depth circuits. The lemma, both as a blackbox and the technical ideas therein are important for our proofs.

Lemma 4.9 (Lemma 5.2 and Lemma 5.3 in [CKS19])**.** *Let $P \in \mathbb{F}[\mathbf{x}, y]$ and let $R(\mathbf{x})$ be polynomials such that R is of degree at most d, $P(\mathbf{x}, R(\mathbf{x})) \equiv 0$ and $\frac{\partial P}{\partial y}(\mathbf{0}, R(\mathbf{0}))$ is non-zero.*
Then, there exists a $(d+1)$-variate polynomial $Q(\mathbf{z})$ of degree at most d such that

$$R(\mathbf{x}) \equiv Q(h_0(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})) \pmod{\langle \mathbf{x} \rangle^{d+1}},$$

where for every $i \in \{0, 1, \ldots, d\}$, $h_i(\mathbf{x})$ is defined as

$$h_i(\mathbf{x}) := \frac{\partial P}{\partial y^j}(\mathbf{x}, R(\mathbf{0})) - \frac{\partial P}{\partial y^j}(\mathbf{0}, R(\mathbf{0})) \text{ trunc } \langle \mathbf{x} \rangle^{d+1}.$$

For our proofs, we end up invoking Lemma 4.9 in settings where the polynomial P also depends on an additional variable T (whereas R does not). In this case, the quantity $\frac{\partial P}{\partial y}(T, \mathbf{0}, R(\mathbf{0}))$ potentially depends on the variable T. Since our root R does not depend on the variable T, we can set T to

some field constant without disturbing the starting conditions for Newton iteration, and thus we can perform Newton iteration as usual.

Lemma 4.10. *Let $P \in \mathbb{F}[T, \mathbf{x}, y]$ and let $R(\mathbf{x})$ be polynomials such that R is of degree at most d, $P(T, \mathbf{x}, R(\mathbf{x})) \equiv 0$ and $\frac{\partial P}{\partial y}(T, \mathbf{0}, R(\mathbf{0}))$ is non-zero.*

Then, there exists a $\kappa \in \mathbb{F}$ and a $(d+1)$-variate polynomial $Q(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$ of degree at most d such that

$$R(\mathbf{x}) \equiv Q(h_0(\kappa, \mathbf{x}), h_1(\kappa, \mathbf{x}), \ldots, h_d(\kappa, \mathbf{x})) \mod \langle \mathbf{x} \rangle^{d+1},$$

where for every $i \in \{0, 1, \ldots, d\}$, $h_i(\mathbf{x})$ is defined as

$$h_i(T, \mathbf{x}) := \frac{\partial P}{\partial y^i}(T, \mathbf{x}, R(\mathbf{0})) - \frac{\partial P}{\partial y^i}(T, \mathbf{0}, R(\mathbf{0})) \text{ trunc } \langle \mathbf{x} \rangle^{d+1}.$$

Proof. Since $\frac{\partial P}{\partial y}(T, \mathbf{0}, R(\mathbf{0})) = \delta(T) \in \mathbb{F}[T]$ for some $\delta(T) \not\equiv 0$, there exists a κ such that $\delta(\kappa) = \delta_0 \neq 0$. Thus, the polynomial $\tilde{P}(\mathbf{x}, y) := P(\kappa, \mathbf{x}, y)$ satisfies $\tilde{P}(\mathbf{x}, R(\mathbf{x})) \equiv 0$ and $\frac{\partial \tilde{P}}{\partial y}(\mathbf{0}, R(\mathbf{0})) = \delta_0 \neq 0$, for some $\delta_0 \in \mathbb{F}$. Applying Lemma 4.9 on $\tilde{P}$ and $R(\mathbf{x})$ tells us that there exists a $(d+1)$-variate polynomial $Q(\mathbf{z})$ of degree at most d such that

$$R(\mathbf{x}) \equiv Q(h_0(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_d(\mathbf{x})) \mod \langle \mathbf{x} \rangle^{d+1},$$

where for every $i \in \{0, 1, \ldots, d\}$, $h_i(\mathbf{x})$ is defined as

$$h_i(\mathbf{x}) := \frac{\partial \tilde{P}}{\partial y^j}(\mathbf{x}, R(\mathbf{0})) - \frac{\partial \tilde{P}}{\partial y^j}(\mathbf{0}, R(\mathbf{0})) \text{ trunc } \langle \mathbf{x} \rangle^{d+1}.$$

Since $\tilde{P}(\mathbf{x}, y) := P(\kappa, \mathbf{x}, y)$, the required statement follows. $\square$

For most of this paper, the only operation we are allowed on the T-variable is scaling by a field element since we care about roots $\mod T^k$, and we want to preserve the T-degree of monomials during any such operations/substitutions. But the above lemma will be invoked at a point when we are concerned about roots $\mod \langle \mathbf{x} \rangle^d$, not $\mod T^k$, which is why it will be okay to replace T by a field element κ.

4.5 Deterministic factorization

For our proof, we need the following classical theorem of Lenstra, Lenstra and Lovasz.

Theorem 4.11 (Factorizing polynomials with rational coefficients [LLL82, vzGG13]). *Let $P \in \mathbb{Q}[x]$ be a monic polynomial of degree d. Then there is a deterministic algorithm computing all the irreducible factors of P that runs in time $\text{poly}(d, t)$, where t is the maximum bit-complexity of the coefficients of f.*

For our proofs, we also rely on a deterministic algorithm for factoring n variate degree d polynomials that run in time $d^{O(n)}$. This is implicit in many known algorithms for polynomial factorization, for instance, in the results of Kopparty, Saraf, Shpilka [KSS15]. Such a statement can also be inferred from our proofs in this paper. We recall a formal statement of this nature from a work of Lecerf below.

Theorem 4.12 ([Lec07, Proposition 4]). *Suppose $P(x_1, \ldots, x_n, y) \in \mathbb{Q}[\mathbf{x}, y]$ be a polynomial that is monic in y with total degree d. Further, suppose that P is squarefree and $P(\mathbf{0}, y)$ is squarefree. Then, there is a deterministic algorithm that takes P as input in the dense representation and outputs each of its irreducible factors in time $\leq O(N^2)$, where $N = \binom{n+d+1}{n}$.*

Intuitively, the proof of the theorem follows from standard Hensel Lifting or Newton Iteration based algorithms for polynomial factorization, and observing (as formally done in [KSS15]) that at every stage of the algorithm, randomness is needed only for polynomial identity testing. Moreover, these PIT instances are all polynomials of degree $\text{poly}(d)$ and on $O(n)$ variables, and hence by Lemma 3.2 can be solved deterministically in time $d^{O(n)}$.

5 Algorithm

5.1 Main theorems and high-level overview

First, we describe our main structural result, which states that the Kabanets-Impagliazzo hitting-set generator, when instantiated with a sufficiently hard low-degree polynomial and an appropriate combinatorial design, preserves the irreducibility of the factors of constant-depth circuits. The precise statement requires a few more conditions, and these conditions are without loss of generality.

Throughout the paper, we will use $\mathcal{G} = \{g_m\}_{m \in \mathbb{N}}$ to denote a family of explicit polynomials such that for every $m \in \mathbb{N}$, $g_m \in \mathbb{F}[x_1, \ldots, x_m]$, $d_m := \deg(g_m) \leq O(\log \log(m))$. Further, $\mathcal{G}$ has the property that for any depth $\Delta \in \mathbb{N}$, if $\mathcal{C} = \{C_m\}_{m \in \mathbb{N}}$ is a family of depth-Δ circuits computing $\mathcal{G}$, then $\mathcal{C}$ requires size $m^{d_m^{\exp(-O(\Delta))}}$, which is $m^{\omega(1)}$. Theorem 3.15 gives us such a family of explicit low-degree polynomials that are hard for constant-depth circuits.

Theorem 5.1 (Irreducibility-preserving variable reduction). *Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. For an absolute constant[6] $C_{\Delta, \varepsilon} \in \mathbb{N}$, let $n \in \mathbb{N}, n \geq C_{\Delta, \varepsilon}$ and $\mathbf{x} := (x_1, \ldots, x_n)$. Let $P(T, \mathbf{x}, z)$ be a nonzero polynomial with the following properties.*

- *$P(T, \mathbf{x}, z)$ is computable by a size $s \leq \text{poly}(n)$ and depth Δ circuit.*

- *$P(T, \mathbf{x}, z)$ is monic in z and T-regularized, with $\deg(P) = D \leq \text{poly}(n)$.*

---

[6] If $a(n) = O(b(n))$, then there exists some C such that for all $n > C$, $a(n) \leq b(n)$; the $C_{\Delta, \varepsilon}$ in our statement is for this purpose. The precise value of $C_{\Delta, \varepsilon}$ depends on the exact hardness of the polynomial in $\mathcal{G}$ and the upper bounds obtained in our proofs.

- $P(T, \mathbf{x}, z)$ and $P(0, \mathbf{x}, z) = P(0, \mathbf{0}, z)$ are squarefree.

Let $\sigma = O(n^\varepsilon)$, $\mu = O(\frac{n^{2\varepsilon}}{\log(n)})$, $\rho = O(\log(n))$, and let $\mathcal{S}$ be an (n, σ, μ, ρ)-design. Let $\mathbf{KI}_{g_\sigma, \mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ be the polynomial map in Definition 3.14 defined using the design $\mathcal{S}$ and the polynomial g_σ from the family of hard polynomials $\mathcal{G}$. Then, the following is true.

A polynomial $F(T, \mathbf{x}, z)$ is an irreducible factor of $P(T, \mathbf{x}, z)$ if and only if $F(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$ is an irreducible factor of $P(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$.

A few remarks regarding the choice of parameters in Theorem 5.1:

- The family $\mathcal{G}$ has degree $d_m \le O(\log\log(m))$ so that $(\log(m))^{\mathrm{poly}(d_m)}$ is $\mathrm{poly}(m)$. We can work with any $d_m \le O((\log m)^\alpha)$ for some small enough α depending on the exponent in $\mathrm{poly}(d_m)$, but $\log\log(m)$ works in every case and makes it simpler to state the theorems.

- The design $\mathcal{S}$ has $\sigma = O(n^\varepsilon)$ because $\mathcal{G}$ is guaranteed to be superpolynomially hard for constant-depth circuits. Stronger hardness guarantees can be used with smaller σ to get algorithms with better time complexity, as is usually the case in hardness-vs-randomness results.

We now describe our main algorithmic result. Informally, the result states that for every choice of depth $\Delta \in \mathbb{N}$, there is a *deterministic* algorithm $\mathcal{A}_\Delta$ such that $\mathcal{A}_\Delta$ takes a depth-Δ circuit for a polynomial P as input, and outputs small (but unbounded depth) circuits along with multiplicity information for each irreducible factor of P. Moreover, $\mathcal{A}_\Delta$ runs in time subexponential in the input size.

Theorem 5.2 (Deterministic subexponential time algorithm for factorization of constant-depth circuits). *Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. There exists an algorithm $\mathcal{A}_{\Delta, \varepsilon}$ which, for all sufficiently large n,*

- *takes as input a polynomial $P(\mathbf{x}) \in \mathbb{Q}[x_1, \ldots, x_n]$ of degree $D \le \mathrm{poly}(n)$ with a depth-Δ, size $s \le \mathrm{poly}(n)$ circuit;*

- *outputs $\mathrm{poly}(s, D)$-sized circuits for each irreducible factor of P, along with the multiplicity of each such factor; and*

- *runs in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$.*

5.1.1 High-level overview of the algorithm

1. Suppose $P(\mathbf{x})$ is the polynomial that we would like to factor. We first use the algorithm by Andrews and Wigderson (Theorem 4.4) to compute the squarefree decomposition of P. Now, we deal with each squarefree part separately.

2. For a specific squarefree part $P_r(\mathbf{x})$, we perform some of the standard transformations ($x_i \mapsto T \cdot x_i + a_i \cdot z + b_i$) so that the polynomial $P_r(T, \mathbf{x}, z)$ is monic in z variable, T-regularized, and $P_r(0, \mathbf{x}, z) = P_r(0, \mathbf{0}, z)$ is squarefree.

3. We apply $\mathbf{KI}_{g,\mathcal{S}}(\mathbf{w})$ on $P_r(T, \mathbf{x}, z)$ (instantiated with an appropriately chosen low-degree hard polynomial g and a design $\mathcal{S}$). This maintains the irreducibility of each factor (Theorem 5.1). At this point, we are working with an n^ε-variate polynomial for some $\varepsilon \in (0, 1)$. Thus, we can use a brute-force / dense-representation factorization algorithm to factorize $P_r(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ in subexponential time (Theorem 4.12).

4. Since $P_r(0, \mathbf{x}, z) = P_r(0, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z) = P_r(0, \mathbf{0}, z)$, taking an irreducible factor of $P_r(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ and setting T to 0 precisely tells us $G(0, \mathbf{x}, z)$ for each irreducible factor $G(T, \mathbf{x}, z)$ of $P_r(T, \mathbf{x}, z)$. Thus, we now have $G(0, \mathbf{x}, z) = G(0, \mathbf{0}, z)$ for each G that is an irreducible factor of P, and we can deal with each G separately.[7]

5. Suppose $G(T, \mathbf{x}, z)$ is an irreducible factor, and we have access to the univariate $G(0, \mathbf{x}, z) = G(0, \mathbf{0}, z)$. We factorize $G(0, \mathbf{0}, z)$ using Theorem 4.11. $G(0, \mathbf{0}, z)$ could have a linear factor of the form $(z - \alpha)$; in this case, we use Newton iteration (Corollary 4.6) to lift the root α to a truncated, approximate z-root $\varphi(T, \mathbf{x})$ of $P(T, \mathbf{x}, z)$ of sufficiently high accuracy. But all the factors of $G(0, \mathbf{0}, z)$ might be non-linear. In this situation, we artificially *add a root u* of $G(0, \mathbf{0}, z)$ to the field; more precisely, if $H(z)$ is an arbitrary irreducible factor of $G(0, \mathbf{0}, z)$, we work over the field $\frac{\mathbb{Q}[u]}{H(u)}$ so that u is now a root of $G(0, \mathbf{0}, z)$. We can efficiently simulate arithmetic in this field. Thus, we can lift the root u to a truncated, approximate z-root $\varphi(T, \mathbf{x})$ of $P(T, \mathbf{x}, z)$ of sufficiently high accuracy.

6. (Lemma 9.1) Given a truncated, approximate z-root φ of $P(T, \mathbf{x}, z)$, we set up a linear system whose solution will give us a minimal polynomial of the root φ. Since we know $G(0, \mathbf{x}, z)$, we know the z-degree of G; this is essentially the information we need to set up the linear system in a way that ensures that the solution is the same as the irreducible factor G.

7. (Theorem 9.2) The solution to the linear system can be represented as a small circuit using Cramer's rule, but this involves a division by a determinant. We shall now use Strassen's division elimination to represent the solution as an arithmetic circuit without division gates; this requires a point where the denominator evaluates to a nonzero value. To this end, we again compose the denominator with $\mathbf{KI}_{g,\mathcal{S}}$ to get an n^ε-variate polynomial; since this denominator essentially captures the uniqueness of the minimal polynomial, we can prove

---

[7]At this point, with the right univariate projections $G(0, \mathbf{0}, z)$ for each irreducible factor $G(T, \mathbf{x}, z)$, one way to get the irreducible factors is to perform *Hensel lifting* (for instance, see [Sud98, Lecture 7] or [ST20]). While Hensel lifting usually ends with a *reconstruction step* that involves solving a linear system, the clean-up for us would just be a truncation since Hensel lifting guarantees that if we start off with the right univariate projection, we will retrieve the right factor, modulo higher degree terms that might have accumulated in the process.

that $\mathbf{KI}_{g,\mathcal{S}}$, by maintaining irreducibility of factors, also maintains the non-zeroness of the denominator. We can now use a brute-force derandomization of the Polynomial Identity Lemma to find a point where the denominator evaluates to a nonzero value, and hence, carry out Strassen's division elimination. Thus, we have division-free circuits for each irreducible factor $G(T, \mathbf{x}, z)$ of $P(T, \mathbf{x}, z)$.

8. Finally, we can undo our initial transformations and output circuits for the irreducible factors of the input polynomial $P(\mathbf{x})$, along with their multiplicities.

5.2 The algorithm

Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. The following algorithm is the "outermost" wrapper for our complete algorithm, and it can be thought of as the algorithm $\mathcal{A}_{\Delta, \varepsilon}$ in Theorem 5.2. The algorithm computes the squarefree decomposition of the input polynomial, and then uses Algorithm 2 on each squarefree part of the decomposition.

Algorithm 1: All factors of depth-Δ circuits, parameter $\varepsilon \in (0, 0.5)$

Input	: A depth-Δ circuit $C_P(\mathbf{x})$ of size $s \in \mathbb{N}$, computing a degree-D polynomial $P(\mathbf{x}) = \prod_{i=1}^{m} G_i(\mathbf{x})^{e_i} \in \mathbb{Q}[\mathbf{x}]$, where $\mathbf{x} = (x_1, \ldots, x_n)$.
Output	: A list $L = \{(C_{G_1}(\mathbf{x}), e_1), \ldots, (C_{G_m}(\mathbf{x}), e_m)\}$, such that each $C_{G_i}(\mathbf{x})$ is a circuit of size poly(s, D) and depth poly(D), which computes the irreducible factor $G_i(\mathbf{x})$, and e_i is the multiplicity of G_i in P.

1 Run Andrews-Wigderson (Theorem 4.4) on $C_P(\mathbf{x})$ to get circuits $C_{P_1}(\mathbf{x}), \ldots, C_{P_r}(\mathbf{x})$ of depth $\Delta' = \Delta + O(1)$, computing $P_1(\mathbf{x}), \ldots, P_r(\mathbf{x})$ where $r = \max_{i \in [m]} e_i$ and $P_i(\mathbf{x}) = \prod_{j \in S_i} G_j(\mathbf{x})$ for $S_i = \{j \in [m] : e_j = i\}$
2 Initialize $L \leftarrow \varnothing$
3 **forall** $i \in [r]$ **do**
4   Run Algorithm 2 for parameters Δ' and ε on $C_{P_i}(\mathbf{x})$ to get a list $L_i := \{C_{G_j}(\mathbf{x}) : j \in S_i\}$
5   For each $C_{G_j}(\mathbf{x}) \in L_i$, add $(C_{G_j}(\mathbf{x}), i)$ to L.

6 **return** L.

5.3 Algorithm for factors of squarefree polynomials

In this section, we will describe the algorithm for steps 3 and 4 of the overview in Section 5.1.1. Fix any $\Delta \in \mathbb{N}$.

Let $\mathcal{G} = \{g_m\}_{m \in \mathbb{N}}$ be a family of polynomials such that for every $m \in \mathbb{N}$, $g_m \in \mathbb{F}[x_1, \ldots, x_m]$, $d_m := \deg(g_m) \leq O(\log \log(m))$. Further, $\mathcal{G}$ has the property that for any depth $\Delta \in \mathbb{N}$, if $\mathcal{C} = \{C_m\}_{m \in \mathbb{N}}$ is a family of depth-Δ circuits computing $\mathcal{G}$, then $\mathcal{C}$ requires size $m^{d_m^{\exp(-O(\Delta))}}$, which is $m^{\omega(1)}$. Theorem 3.15 gives us such a family of explicit low-degree polynomials that are hard for constant-depth circuits.

Let $\varepsilon \in (0, 0.5)$. Let $\sigma = O(n^\varepsilon)$, $\mu = O(\frac{n^{2\varepsilon}}{\log(n)})$, $\rho = O(\log(n))$, and let $\mathcal{S}$ be an (n, σ, μ, ρ)-design. Let $\mathbf{KI}_{g_\sigma, \mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ be the polynomial map in Definition 3.14 defined using the design $\mathcal{S}$ and the polynomial g_σ from the family of hard polynomials $\mathcal{G}$.

Algorithm 2: Factors of squarefree polynomial with depth-Δ circuits, parameter $\varepsilon \in (0, 0.5)$

    Input   **:** A depth-Δ size-s circuit C_P computing the squarefree degree-D polynomial
                        $P(\mathbf{x}) = \prod_{j \in [m]} G_j(\mathbf{x})$, where the G_js are the irreducible factors of P.

    Output   **:** A list $L = \{C_{G_1}(\mathbf{x}), \ldots, C_{G_m}(\mathbf{x})\}$, such that each $C_{G_j}(\mathbf{x})$ is a circuit of size $\mathrm{poly}(s, D)$
                        and depth $\mathrm{poly}(D)$, which computes the irreducible factor $G_j(\mathbf{x})$.

1 Compute $\mathbf{a} \in \mathbb{Q}^n$ such that $\delta = \mathrm{Hom}_D[P](\mathbf{a}) \neq 0$ using Theorem 3.16. Let $\hat{P}(\mathbf{x}, z) := P(\mathbf{x} + (\mathbf{a} \cdot z))/\delta$
   and for each $j \in [m]$, $\hat{G}_j(T, \mathbf{x}, z) := G_j(\mathbf{x} + (\mathbf{a} \cdot z))/\mathrm{Hom}_{\deg(G_j)}[G_j](\mathbf{a})$
   `// Ensures that` $\hat{P}$ `is monic in z`

2 Compute $\mathbf{b} \in \mathbb{Q}^n$ such that $\mathrm{Disc}_z(\hat{P})(\mathbf{b}) \neq 0$ using Theorem 3.16. Let $\tilde{P}(T, \mathbf{x}, z) := \hat{P}((T \cdot \mathbf{x}) + \mathbf{b}, z)$
   and for each $j \in [m]$, $\tilde{G}_j(T, \mathbf{x}, z) := \hat{G}_j((T \cdot \mathbf{x}) + \mathbf{b}, z)$
   `// Ensures that` $\tilde{P}(T, \mathbf{x}, z)$ `is T-regularised, and` $P(0, 0, z)$ `is squarefree.`

3 Construct the map $\mathbf{KI}_{g_\sigma, \mathcal{S}}$ using Lemma 3.13 and $g_\sigma \in \mathcal{G}$ as given by Theorem 3.15.

4 Factorize $C_{\tilde{P}}(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ so that for each $j \in [m]$, $C_{\tilde{G}_j}(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ is a circuit that computes
   $\tilde{G}_j(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$, satisfying the property that $\tilde{G}_j(0, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z) = \tilde{G}_j(0, \mathbf{0}, z)$.

5 Run Algorithm 3 with parameters $\Delta + 2$ and ε on $C_{\tilde{P}}(T, \mathbf{x}, z)$ and $\{\tilde{G}_j(0, \mathbf{0}, z) : j \in [m]\}$ to get a list
   $L' = \{C_{\tilde{G}_1}(T, \mathbf{x}, z), \ldots, C_{\tilde{G}_m}(T, \mathbf{x}, z)\}$.

6 For each $C_{\tilde{G}_j}(T, \mathbf{x}, z) \in L'$, compute $C_{G_j}(\mathbf{x}) := C_{\tilde{G}_j}(1, \mathbf{x} - \mathbf{b}, 0)$.

7 Output $L = \{C_{G_1}(\mathbf{x}), \ldots, C_{G_m}(\mathbf{x})\}$

5.4 Algorithm for obtaining irreducible factors from the right univariate projections

In this section, we describe the algorithm for steps 5-7 of the overview in Section 5.1.1. Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$.

Algorithm 3: Factors of monic, T-regularized, squarefree polynomial with depth-Δ circuit, parameter $\varepsilon \in (0, 0.5)$

Input :
- A depth-Δ size-s circuit $C_{\tilde{P}}$ computing a squarefree degree-D polynomial $\tilde{P}(T, \mathbf{x}, z) = \prod_{j \in [m]} \tilde{G}_j(T, \mathbf{x}, z)$, where the $\tilde{G}_i$s are the irreducible factors of $\tilde{P}$. Further, $\tilde{P}(T, \mathbf{x}, z)$ is monic in z, T-regularized with respect to $\mathbf{x}$ and $C_{\tilde{P}}(0, \mathbf{x}, z) = C_{\tilde{P}}(0, \mathbf{0}, z)$ is squarefree.

- For each $j \in [m]$, the univariate polynomial $\tilde{G}_j(0, \mathbf{x}, z) = \tilde{G}_j(0, \mathbf{0}, z)$.

Output : A list $L = \{C_{\tilde{G}_1}(T, \mathbf{x}, z), \ldots, C_{\tilde{G}_m}(T, \mathbf{x}, z)\}$, such that each $C_{\tilde{G}_i}(T, \mathbf{x}, z)$ is a circuit of size poly(s, D) and depth poly(D), which computes the irreducible factor $\tilde{G}_i(T, \mathbf{x}, z)$ of $\tilde{P}(T, \mathbf{x}, z)$.

Prerequisites : $\mathbf{KI}_{g_\sigma, \mathcal{G}}$ constructed in Algorithm 2 using Lemma 3.13 and Theorem 3.15

---

1 Initialize $L \leftarrow \varnothing$

2 **forall** $j \in [m]$ **do**

3 $\quad$ Factorize the degree-D_z polynomial $\tilde{G}_j(0, \mathbf{0}, z)$ (Theorem 4.11) to get $\tilde{G}_j(0, \mathbf{0}, z) = \prod_{l=1}^{r_j} H_l(z)$, where every $H_l(z)$ is distinct, monic and irreducible.

4 $\quad$ Let $H(z)$ be an arbitrary irreducible factor of $\tilde{G}_j(0, \mathbf{0}, z)$. Let $\mathbb{K}$ be the field $\frac{\mathbb{Q}[u]}{H(u)}$.

5 $\quad$ Use Newton iteration (Lemma 4.5) to compute a truncated, non-degenerate, approximate z-root $\varphi(T, \mathbf{x}) \in \mathbb{K}[T, \mathbf{x}]$ of $\tilde{G}_j(T, \mathbf{x}, z)$ of order $2D \cdot D_z + 1$ such that $\varphi(0, \mathbf{x}) = \varphi(0, \mathbf{0}) = u \in \mathbb{K}$.

6 $\quad$ Use the algorithm from Theorem 9.2 (with $\mathbf{KI}_{g_\sigma, \mathcal{G}}$) on $\varphi(T, \mathbf{x})$ to compute a circuit $C_{\tilde{G}_j}$ for $\tilde{G}_j$.

7 $\quad$ Add $C_{\tilde{G}_j}$ to L.

8 **return** L

---

6 Technical building blocks

6.1 Properties of roots modulo T^k

The following lemma observes that certain homomorphisms defined by polynomial maps preserve approximate roots for a polynomial. This will be used to argue that an approximate root remains an approximate root (with some nice properties) even after we plug in the Kabanets-Impagliazzo generator.

Lemma 6.1 (Preserving root properties under homomorphisms). *Let $k > 1$ be any natural number and let $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ and $R(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials satisfying the following properties.*

- *$P(T, \mathbf{x}, z)$ is T-regularized and it is monic in z.*

- *$R(T, \mathbf{x})$ is a truncated, non-degenerate, approximate z-root of $P(T, \mathbf{x}, y)$ of order k.*

For a new tuple $\mathbf{w}$ of μ variables distinct from $T, \mathbf{x}, z$, polynomials $h_1, h_2, \ldots, h_n \in \mathbb{F}[\mathbf{w}]$ and a non-zero field constant $\gamma \in \mathbb{F}$, let $\Lambda : \mathbb{F}[T, \mathbf{x}, z] \to \mathbb{F}[T, \mathbf{w}, z]$ be the ring homomorphism defined by $T \mapsto \gamma T$, $x_i \mapsto h_i(\mathbf{w})$ and $z \mapsto z$. Then, the following are true.

- $\Lambda(R)(0, \mathbf{w}) = R(0, \mathbf{x}) \in \mathbb{F}$

- $\Lambda(P)(T, \mathbf{w}, z)$ *is T-regularized, and it is monic in z.*

- $\Lambda(R)(T, \mathbf{w})$ *is a truncated, non-degenerate, approximate z-root of $\Lambda(P)(T, \mathbf{w}, y)$ of order k*

Proof. The lemma follows essentially because Λ is a homomorphism.

1. By Observation 4.2, $R(0, \mathbf{x}) \in \mathbb{F}$. Since Λ is a homomorphism with the property that $\deg_T(f) = \deg_T(\Lambda(f))$ for any $f \in \mathbb{F}[T, \mathbf{x}, z]$, it follows that $\Lambda(R)(0, \mathbf{w}) = R(0, \mathbf{x})$.

2. If a monomial $\mathbf{m} \in \mathbb{F}[T, \mathbf{x}, z]$, is T-regularized, then so is $\Lambda(\mathbf{m})$. Since Λ is a homomorphism, this property extends to all polynomials. Similarly, since $\Lambda(z) = z$, $\Lambda(P)$ remains monic in z.

3. $P(T, \mathbf{x}, R(T, \mathbf{x})) \equiv 0 \mod T^k$, or equivalently, every monomial in $P(T, \mathbf{x}, R(T, \mathbf{x}))$ has T-degree at least k. As observed in the first point, Λ is a homomorphism with the property that $\deg_T(f) = \deg_T(\Lambda(f))$ for any $f \in \mathbb{F}[T, \mathbf{x}, z]$. Thus, $\Lambda(P)(T, \mathbf{w}, \Lambda(R)(T, \mathbf{w})) \equiv 0 \mod T^k$.

$\square$

The following lemma observes that truncated, non-degenerate, approximate roots of a polynomial are unique once the value of the root at $T = 0$ is decided; it follows almost immediately from the uniqueness of approximate roots computed via Newton iteration.

Lemma 6.2 (Uniqueness of truncated, non-degenerate, approximate roots). *Let $k \in \mathbb{N}$. Suppose $\Psi_1(T, \mathbf{x})$ and $\Psi_2(T, \mathbf{x})$ are truncated, non-degenerate, approximate z-roots of order k with respect to T for a T-regularized polynomial $P(T, \mathbf{x}, z)$. If $\Psi_1(0, \mathbf{x}) = \Psi_2(0, \mathbf{x}) = \alpha \in \mathbb{F}$, then $\Psi_1(T, \mathbf{x}) \equiv \Psi_2(T, \mathbf{x})$.*

Proof. Since $\Psi_1(T, \mathbf{x})$ is a non-degenerate root of $P(T, \mathbf{x}, z)$ which is T-regularized, $\frac{\partial P}{\partial z}(0, \mathbf{x}, \Psi_1(0, \mathbf{x})) = \frac{\partial P}{\partial z}(0, \mathbf{x}, \Psi_1(0, \mathbf{x})) = \frac{\partial P}{\partial z}(0, \mathbf{0}, \alpha) = \beta \in \mathbb{F}$ is nonzero. Similarly, $\frac{\partial P}{\partial z}(0, \mathbf{x}, \Psi_2(0, \mathbf{x})) = \frac{\partial P}{\partial z}(0, \mathbf{x}, \Psi_2(0, \mathbf{x})) = \frac{\partial P}{\partial z}(0, \mathbf{0}, \alpha) = \beta$. Since α satisfies $P(T, \mathbf{x}, \alpha) \equiv 0 \mod T$ and $\frac{\partial P}{\partial z}(0, \mathbf{x}, \alpha) = \beta \in \mathbb{F}$ for $\beta \neq 0$, Lemma 4.5 tells us that there is a unique truncated, approximate z-root $\Phi(T, \mathbf{x})$ of order k for $P(T, \mathbf{x}, z)$, satisfying $\Phi(0, \mathbf{x}) = \alpha$. Thus, $\Phi(T, \mathbf{x}) \equiv \Psi_1(T, \mathbf{x}) \equiv \Psi_2(T, \mathbf{x})$. $\square$

6.2 Complexity of low degree homogeneous components of roots

The following is our main technical lemma where we argue that the low-degree homogeneous components of an approximate root for a constant-depth circuit can be computed by a small constant-depth circuit.

Lemma 6.3. *Let $k > 1$ be any natural number and let $P(T, \mathbf{w}, z) \in \mathbb{F}[T, \mathbf{w}, z]$ and $R(T, \mathbf{w}) \in \mathbb{F}[T, \mathbf{w}]$ be polynomials satisfying the following properties.*

- $P(T, \mathbf{w}, z)$ *is computable by a depth Δ circuit of size s.*

- $P(T, \mathbf{w}, z)$ is T-regularized and monic in z.

- $R(T, \mathbf{w})$ is a truncated, non-degenerate, approximate z-root of $P(T, \mathbf{x}, z)$ of order k with respect to T.

Then, for every $\ell \in \mathbb{N}$, there is an algebraic circuit $C_\ell \in \mathbb{F}[T, \mathbf{w}]$ of depth at most $(\Delta + O(1))$, size at most $\left(\operatorname{poly}(s, \deg(P)) \cdot (\log k)^{\operatorname{poly}(\ell)} \right)$

$$C_\ell(T, \mathbf{w}) = R(T, \mathbf{w}) \text{ trunc } \langle \mathbf{w} \rangle^\ell.$$

Proof. As seen in Observation 4.2, $R(0, \mathbf{w})$ is a root of $P(0, \mathbf{0}, z)$, and hence must be a field element α (possibly from an extension of $\mathbb{F}$) and does not depend on $\mathbf{w}$. Similarly, we also get that $\frac{\partial P}{\partial z}(0, \mathbf{w}, R(0, \mathbf{w}))$ is a non-zero field element that we denote by β. Thus, $R(T, \mathbf{w})$ is the *unique* lift of the root α of $P(T, \mathbf{w}, z)$ modulo T to a root modulo the ideal T^k (Lemma 6.2). In particular, we can view R as an outcome of Newton Iteration (and then eventual truncation modulo T^k). We prove the lemma by induction on this iteration, and maintaining the following inductive claim. Let $2^m \in [k, 2k]$ be the smallest power of 2 greater than k. From Lemma 4.8, we get that the sequence of polynomials $\varphi_0, \varphi_1, \ldots, \varphi_m, \psi_0, \psi_1, \ldots, \psi_m \in \mathbb{F}[T, \mathbf{w}]$ defined as

$$\varphi_0 = \alpha, \qquad\qquad\qquad \psi_0 = (1/\beta),$$

$$\text{For } i \geq 0, \quad \varphi_{i+1} = \varphi_i - P(T, \mathbf{w}, \varphi_i) \cdot \psi_i, \qquad \psi_{i+1} = 2\psi_i - \psi_i^2 \cdot \frac{\partial P}{\partial z}(T, \mathbf{w}, \varphi_{i+1}).$$

satisfy

$$R(T, \mathbf{w}) = \varphi_m(T, \mathbf{w}) \text{ trunc } T^k.$$

We note that while we are dealing with polynomials in both T and $\mathbf{w}$ variables, the lifting is happening only with respect to T. In this sense, we are really viewing the polynomial $P(T, \mathbf{w}, z)$ as a polynomial in $\mathbb{F}[\mathbf{w}][T, z]$ for the purpose of this lifting. We now use the following claim, whose proof we defer to the end of this section, to complete the proof of the lemma.

Claim 6.4. *For every $i \geq 0$, there exists a set $\mathcal{G}_i$ of at most $\tau_i \leq 2(\ell + 1)i$ polynomials $\{g_1, g_2, \ldots, g_{\tau_i}\}$ in $\mathbb{F}[T, \mathbf{w}]$ and two $(\tau_i + 1)$-variate polynomials $Q_i(T, u_1, \ldots, u_{\tau_i}), \hat{Q}_i(T, u_1, \ldots, u_{\tau_i}) \in \mathbb{F}[T, \mathbf{u}]$ such that*

$$\varphi_i \equiv Q_i(T, g_1, \ldots, g_{\tau_i}) \mod \langle \mathbf{w} \rangle^\ell,$$

and

$$\psi_i \equiv \hat{Q}_i(T, g_1, \ldots, g_{\tau_i}) \mod \langle \mathbf{w} \rangle^\ell.$$

Moreover, each polynomial g in $\mathcal{G}_i$ is of the form $\frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma)$ for some $j \leq (\ell + 1)$ and $\gamma \in \mathbb{F}[T]$.

From the fact that $R(T, \mathbf{w}) = \varphi_m(T, \mathbf{w})$ trunc T^k, we get that

$$R(T, \mathbf{w}) \text{ trunc } \langle \mathbf{w} \rangle^\ell = \left(\varphi_m(T, \mathbf{w}) \text{ trunc } T^k \right) \text{ trunc } \langle \mathbf{w} \rangle^\ell.$$

Now, since T and $\mathbf{w}$ are disjoint variables, we can exchange the order of the operations of truncating modulo T and truncating modulo $\langle \mathbf{w} \rangle^\ell$. So, we have that

$$R(T, \mathbf{w}) \text{ trunc } \langle \mathbf{w} \rangle^\ell = \left(\varphi_m(T, \mathbf{w}) \text{ trunc } \langle \mathbf{w} \rangle^\ell \right) \text{ trunc } T^k.$$

From Claim 6.4, we get that

$$R(T, \mathbf{w}) \text{ trunc } \langle \mathbf{w} \rangle^\ell = \left(Q_m(T, g_1, g_2, \ldots, g_{\tau_m}) \text{ trunc } \langle \mathbf{w} \rangle^\ell \right) \text{ trunc } T^k.$$

We would like to show that the RHS of the above equation has a small constant-depth circuit. Since we eventually truncate to T-degree $k - 1$, we can assume without loss of generality that the T-degree of Q_m and each g_i is at most $k - 1$. In particular, if g_i equals $\frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma)$ for some $j \in \mathbb{N}$ and $\gamma \in \mathbb{F}[T]$, this γ can be assumed to be of T-degree at most $(k - 1)$. Moreover, since $P(T, \mathbf{w}, z)$ is a polynomial with a depth-Δ circuit of size s, we get, from Corollary 3.4 that $\frac{\partial P}{\partial z^j}(T, \mathbf{w}, z)$ has a depth $(\Delta + O(1))$ circuit of size at most $s \cdot \mathrm{poly}(\deg(P))$. From the bound on the degree of $\gamma \in \mathbb{F}[T]$, we get that each $g_i(T, \mathbf{w})$ has a depth $(\Delta + O(1))$ circuit of size at most $s \cdot \mathrm{poly}(\deg(P))$.

Finally, we note that we can view $Q_m(T, g_1, g_2, \ldots, g_{\tau_m})$ as

$$\tilde{Q}_m(T, g_1(T, \mathbf{w}) - g_1(T, \mathbf{0}), g_2(T, \mathbf{w}) - g_2(T, \mathbf{0}), \ldots, g_{\tau_m}(T, \mathbf{w}) - g_{\tau_m}(T, \mathbf{0})),$$

for some polynomial $\tilde{Q}_m$. Since we are only interested in working with the above polynomial modulo $\langle \mathbf{w} \rangle^\ell$ and every monomial in each of the polynomials $g_j(T, \mathbf{w}) - g_j(T, \mathbf{0})$ has $\mathbf{w}$-degree at least one, we get that $\tilde{Q}_m(T, u_1, \ldots, u_{\tau_m})$ can be assumed to have degree at most ℓ in the $\mathbf{u}$ variables. Thus, $\tilde{Q}_m(T, u_1, \ldots, u_{\tau_m})$ can be computed by a depth-2 circuit of size at most $k \cdot \binom{\tau_m + \ell}{\ell} \le k \cdot (\log k)^{\mathrm{poly}(\ell)}$. Combining this circuit with the constant-depth circuits for $g_j(T, \mathbf{w})$ (and hence $g_j(T, \mathbf{w}) - g_j(T, \mathbf{0})$) gives us a depth-$(\Delta + O(1))$ circuit $\tilde{C}$ of size at most $\mathrm{poly}(s, \deg(P)) \cdot (\log k)^{\mathrm{poly}(\ell)}$ satisfying

$$\tilde{C}(T, \mathbf{w}) \equiv Q_m(T, g_1, \ldots, g_{\tau_m}) \bmod T^k \bmod \langle \mathbf{w} \rangle^{\ell+1}.$$

The T-degree of this circuit is at most $\deg_T(\tilde{Q}_m) \cdot \ell \cdot k \le \mathrm{poly}(k, \ell)$. The $\mathbf{w}$-degree of this circuit is at most $\ell \cdot \deg(P)$. Thus, we can apply the operations trunc $\langle \mathbf{w} \rangle^{\ell+1}$ and trunc T^k by an application of Corollary 3.4 to get the circuit $C_\ell(T, \mathbf{w})$ in the conclusion of the lemma. $\square$

We now discuss the proof of Claim 6.4.

Proof of Claim 6.4. We prove the claim via an induction on i. For $i = 0$, we already know that $\varphi_0 = \alpha$

and $\psi_0 = \frac{1}{\beta}$ are polynomials in $\mathbb{F}[T]$ and hence the claim immediately holds. We now assume that the claim holds for φ_i and ψ_i i.e.

$$\varphi_i \equiv Q_i(T, g_1, \ldots, g_{\tau_i}) \mod \langle \mathbf{w} \rangle^\ell,$$

and

$$\psi_i \equiv \hat{Q}_i(T, g_1, \ldots, g_{\tau_i}) \mod \langle \mathbf{w} \rangle^\ell$$

and show that it must hold for φ_{i+1} and ψ_{i+1}.

From the inductive definitions of φ_{i+1} and ψ_{i+1}, we get that

$$\varphi_{i+1} \equiv \varphi_i - P(T, \mathbf{w}, \varphi_i) \cdot \psi_i$$

and

$$\psi_{i+1} \equiv 2\psi_i - \psi_i^2 \cdot \frac{\partial P}{\partial z}(T, \mathbf{w}, \varphi_{i+1})$$

If $\gamma_i := \varphi_i(T, \mathbf{0})$, i.e. γ_i is the $\mathbf{w}$-free part of φ_i, we get via a Taylor expansion that

$$P(T, \mathbf{w}, \varphi_i) = P(T, \mathbf{w}, \gamma_i + (\varphi_i - \gamma_i)) = \sum_{j=0}^{\deg(P)} \frac{1}{j!} \cdot \frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma_i) \cdot (\varphi_i - \gamma_i)^j. \tag{6.5}$$

From the definition of γ_i, we have that every monomial in $(\varphi_i - \gamma_i)$ has $\mathbf{w}$-degree at least one. Thus, for every $j \geq \ell$, we have that $(\varphi_i - \gamma_i)^j \equiv 0$ modulo $\langle \mathbf{w} \rangle^\ell$. Thus, we have that

$$P(T, \mathbf{w}, \varphi_i) \equiv \sum_{j=0}^{\ell} \frac{1}{j!} \cdot \frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma_i) \cdot (\varphi_i - \gamma_i)^j \mod \langle \mathbf{w} \rangle^\ell.$$

Similarly,

$$\frac{\partial P}{\partial z}(T, \mathbf{w}, \varphi_{i+1}) \equiv \sum_{j=0}^{\ell} \frac{1}{j!} \cdot \frac{\partial P}{\partial z^{j+1}}(T, \mathbf{w}, \gamma_{i+1}) \cdot (\varphi_{i+1} - \gamma_{i+1})^j \mod \langle \mathbf{w} \rangle^\ell.$$

Thus, φ_{i+1} can be written as a polynomial in $Q_i, \hat{Q}_i$ and polynomials in the set $\{\frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma_i) : j \in \{0, 1, \ldots, \ell\}\}$, where the coefficients of this polynomial are from the ring $\mathbb{F}[T]$ and all equalities hold modulo $\langle \mathbf{w} \rangle^\ell$. Similarly, ψ_{i+1} can be written as a polynomial in $Q_{i+1}, \hat{Q}_i$ and polynomials in the set $\{\frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma_{i+1}) : j \in \{1, 2, \ldots, \ell+1\}\}$, where the coefficients of this polynomial are from the ring $\mathbb{F}[T]$ and all equalities hold modulo $\langle \mathbf{w} \rangle^\ell$. Moreover, in going from i to $(i+1)$, we have increased the size of the *generating set* additively by at most $2(\ell+1)$. Furthermore, each of the new

elements of this generating set is again of the form $\frac{\partial P}{\partial z^j}(T, \mathbf{w}, \gamma)$ for some $j \leq (\ell + 1)$ and $\gamma \in \mathbb{F}[T]$.

$\hfill \square$

An immediate consequence of Lemma 6.3 is the following corollary.

Corollary 6.6. *Let $k \in \mathbb{N}$, $P(T, \mathbf{w}, z) \in \mathbb{F}[T, \mathbf{w}, z]$ and $R(T, \mathbf{w}) \in \mathbb{F}[T, \mathbf{w}]$ be polynomials that satisfy the hypothesis of Lemma 6.3.*

Then, for every $\ell \in \mathbb{N}$, there is an algebraic circuit $C_\ell \in \mathbb{F}[T, \mathbf{w}]$ of depth at most $2\Delta + O(1)$ and size at most $\left(\mathrm{poly}(s, \deg(P)) \cdot (\log k)^{\mathrm{poly}(\ell)} \right)$ such that

$$C_\ell(T, \mathbf{w}) \equiv P(T, \mathbf{w}, R(T, \mathbf{w})) \ \ \mathrm{trunc} \ \langle \mathbf{w} \rangle^\ell .$$

6.2.1 Why do we need the quadratic convergence version of Newton iteration?

It would be instructive to step back and see why Newton iteration with quadratic convergence was required in the above argument. A slight modification of Lemma 4.9 will let us argue that low-degree components of approximate roots of a polynomial have small constant-depth circuits. However, there are subtle differences in the statement of Lemma 4.9 and the statement of Lemma 6.3. The key difference is that the approximate root φ_k is with respect to the variable T, whereas we are extracting homogeneous components of degree up to ℓ with respect to a *different set of variables*. Therefore, we may have contributions of $\mathbf{w}$-degree at most ℓ from terms with T-degree up to k. So, we cannot replace φ_k by a root of lower accuracy such as φ_ℓ.

In the above proof of Lemma 6.3, each level of the iteration adds to the number of "generators" used in the composition. Using the standard Newton iteration of k steps for φ_k would result in $O(k)$ generators and would not yield the required size bounds needed for our proof. The version of Newton iteration with quadratic convergence ensures that we obtain φ_k from $\log k$ iterations, and results in the eventual number of "generators" as $O(\log k)$ instead (which was crucial for the final circuit size bound).

7 Warm-up: preserving true roots under variable reduction

In this section, we use the techniques of Section 6 to show a variable reduction map that will help us identify whether a given truncated power series root of a constant-depth circuit is a true root (and not just a sufficiently good truncation of a power series root). As described in the proof overview, the idea is to work with a PIT instance formed by plugging-in a candidate root into the input polynomial. In Lemma 7.1, we show that low-degree roots of such PIT instances have relatively small constant-depth circuits when the input polynomial has a small constant-depth circuit.

Lemma 7.1. *Let $k > 1$ be any natural number and let $P(T, \mathbf{w}, y, z) \in \mathbb{F}[T, \mathbf{w}, z]$ and $R(T, \mathbf{w}, y) \in \mathbb{F}[T, \mathbf{w}, y]$ be polynomials satisfying the following properties.*

- $P(T, \mathbf{w}, y, z)$ is computable by a depth Δ circuit of size s.

- $P(T, \mathbf{w}, y, z)$ is T-regularized and monic in z.

- $R(T, \mathbf{w}, y)$ is a truncated, non-degenerate, approximate z-root of $P(T, \mathbf{w}, y, z)$ of order k.

Let $F(\mathbf{w}) \in \mathbb{F}[\mathbf{w}]$ be a polynomial of degree at most ℓ such that $(y - F(\mathbf{w}))$ divides $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$. Then, F can be computed by an algebraic circuit of depth $(\Delta + O(1))$ and size at most

$$\mathrm{poly}(s, \deg(P)) \cdot (\log k)^{\mathrm{poly}(\ell)}.$$

Proof of Lemma 7.1. We start by setting ourselves up to invoke Lemma 4.10, a version of Lemma 4.9 from [CKS19]. To do this, we first need to ensure that the hypothesis of the lemma holds in our case.

Setting up to invoke Lemma 4.10: The first issue is that $F(\mathbf{w})$ could be a y-root of high multiplicity of $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$. To work around this, we work with an appropriately high order derivative of P with respect to y. To this end, we start by viewing $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$ as a univariate polynomial in y with coefficients from the ring $\mathbb{F}[T, \mathbf{w}]$. Thus, $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$ can be decomposed as

$$P(T, \mathbf{w}, y, R(T, \mathbf{w}, y)) = \sum_{i=0}^{D} P_i(T, \mathbf{w}) y^i,$$

where each P_i is a polynomial in only T and $\mathbf{w}$ variables. Let $(m + 1)$ be the largest integer such that $(Y - F(\mathbf{w}))^{m+1}$ divides $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$. So, to reduce the multiplicity of the y-root $F(\mathbf{w})$ to 1, we consider the polynomial $\hat{P}$ defined as

$$\hat{P}(T, \mathbf{w}, y) := \frac{\partial P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))}{\partial y^m}.$$

Thus, we have that $(y - F(\mathbf{w}))$ divides $\hat{P}$ and does not divide $\frac{\partial \hat{P}}{\partial y}$. As a consequence, we get that $\frac{\partial \hat{P}}{\partial y}(T, \mathbf{w}, F(\mathbf{w}))$ is a non-zero polynomial in $\mathbb{F}[T, \mathbf{w}]$. By shifting the $\mathbf{w}$ variables if needed to $\mathbf{a} + \mathbf{w}$ for some $\mathbf{a} \in \mathbb{F}^{|\mathbf{w}|}$, we get that

$$\frac{\partial \hat{P}}{\partial y}(T, \mathbf{0}, F(\mathbf{0})) \not\equiv 0.$$

We are now in a position to invoke Lemma 4.10, which tells us that there exists a $\kappa \in \mathbb{F}$ and a $(\ell + 1)$-variate polynomial $Q(\mathbf{u}) \in \mathbb{F}[\mathbf{u}]$ of degree at most ℓ such that

$$F(\mathbf{w}) \equiv Q(h_0(\kappa, \mathbf{w}), h_1(\kappa, \mathbf{w}), \ldots, h_\ell(\kappa, \mathbf{w})) \mod \langle \mathbf{w} \rangle^{\ell+1}, \tag{7.2}$$

where for every $i \in \{0, 1, \ldots, \ell\}$, $h_i(T, \mathbf{w})$ is defined as

$$h_j(T, \mathbf{w}) := \left(\frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{w}, F(\mathbf{0})) - \frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{0}, F(\mathbf{0})) \right) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1}.$$

Given the bounds on the number of variables and $\mathbf{u}$-degree of $Q(\mathbf{u})$, we get that $Q(\mathbf{u})$ is computable by an algebraic circuit of depth-2 and size at most $\exp(O(\ell))$. At this point, if we can somehow replace each $h_i(T, \mathbf{w})$ by a small constant-depth circuit, we can obtain a small constant-depth circuit for $F(\mathbf{w})$ by combining these with the depth-2 circuit for Q, and then extracting certain homogeneous components of interest.

Constant-depth circuits for $h_i(T, \mathbf{w})$: Recall that $\hat{P}(T, \mathbf{w}, y)$ is a derivative of $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$ with respect to y^m. Thus, from Corollary 3.4, we get that $\hat{P}$ can be written as an $\mathbb{F}[y]$-weighted linear combination of the polynomials in the set

$$\{P(T, \mathbf{w}, \beta_i, R(T, \mathbf{w}, \beta_i)) : i \in \{0, 1, \ldots, \deg(P)\}\}$$

where β_i's are distinct field constants and every weight in the linear combination has y-degree at most $\deg(P)$.

$$\begin{aligned}
h_j(T, \mathbf{w}) &= \left(\frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{w}, F(\mathbf{0})) - \frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{0}, F(\mathbf{0})) \right) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1} \\
&= \left(\left(\frac{\partial}{\partial y^{j+m}} P(T, \mathbf{w}, y, R(T, \mathbf{w}, y)) \right) \Big|_{y=F(\mathbf{0})} - \frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{0}, F(\mathbf{0})) \right) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1} \\
&= \left(\left(\sum_{i=0}^{D} \Lambda_i(F(\mathbf{0})) P(T, \mathbf{w}, \beta_i, R(T, \mathbf{w}, \beta_i)) \right) - \frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{0}, F(\mathbf{0})) \right) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1} \\
&= \left(\sum_{i=0}^{D} \Lambda_i(F(\mathbf{0})) P(T, \mathbf{w}, \beta_i, R(T, \mathbf{w}, \beta_i)) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1} \right) - \frac{\partial \hat{P}}{\partial y^j}(T, \mathbf{0}, F(\mathbf{0}))
\end{aligned}$$

Thus, to show that $h_i(T, \mathbf{w})$ have small constant-depth circuits, it suffices (up to multiplicative factors of $\text{poly}(\deg(P))$) to show that the polynomials $(P(T, \mathbf{w}, \beta_i, R(T, \mathbf{w}, \beta_i)) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1})$ have small constant-depth circuits. But, this is exactly the content of Corollary 6.6[8], which shows that $(P(T, \mathbf{w}, \beta_i, R(T, \mathbf{w}, \beta_i)) \text{ trunc } \langle \mathbf{w} \rangle^{\ell+1})$ has a circuit of depth $\Delta + O(1)$ and size

$$\left(\text{poly}(s, \deg(P)) \cdot (\log k)^{\text{poly}(\ell)} \right).$$

This gives a circuit $C_i(T, \mathbf{w})$ of roughly the same size with a constant additive increase in depth for each $h_i(T, \mathbf{w})$, and thus a circuit $\tilde{C}_i(\mathbf{w}) := C_i(\kappa, \mathbf{w})$ for each $h_i(\kappa, \mathbf{w})$, where κ is the field constant

---

[8]There is a slight subtlety here: $P(T, \mathbf{w}, y, R(T, \mathbf{w}, y))$ must still satisfy the hypothesis of Lemma 6.3/Corollary 6.6 after replacing each y by a field constant β_i. But this is true because of Lemma 6.1.

in Equation (7.2).

Putting things together: Plugging in these circuits as inputs to the depth-2 circuit for $Q(\mathbf{u})$, we get that there is a circuit $C(\mathbf{w}) = Q(\tilde{C}_0(\mathbf{w}), \ldots, \tilde{C}_\ell(\mathbf{w}))$ of size $\left(\text{poly}(s, \deg(P)) \cdot (\log k)^{\text{poly}(\ell)}\right)$ and depth $(\Delta + O(1))$ such that

$$F(\mathbf{w}) \equiv C(\mathbf{w}) \bmod \langle \mathbf{w} \rangle^{\ell+1}.$$

The $\mathbf{w}$-degree of $C(\mathbf{w})$ is at most $\ell \cdot \deg(Q)$, which is at most ℓ^2. By using Corollary 3.4 to compute the truncation $C(\mathbf{w})$ trunc $\langle \mathbf{w} \rangle^{\ell+1}$, we get a depth-$(\Delta + O(1))$ circuit for $F(\mathbf{w})$, of size at most

$$\left(\text{poly}(s, \deg(P)) \cdot (\log k)^{\text{poly}(\ell)}\right).$$

$\square$

Now, we will use Lemma 7.1 to prove the main theorem of this section: the Kabanets-Impagliazzo generator, instantiated with an appropriate design and a low-degree polynomial, preserves the roots of constant-depth circuits, while ensuring that no new roots are created. This is a special case of Theorem 5.1.

Theorem 7.3 (Checking validity of a root). *Let $k > 1$ be any natural number and let $A(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ and $\Phi(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials satisfying the following properties.*

- *$A(T, \mathbf{x}, z)$ is computable by a depth Δ circuit of size s.*

- *$A(T, \mathbf{x}, z)$ is T-regularized and monic in z.*

- *$\Phi(T, \mathbf{x})$ is a truncated, non-degenerate, approximate z-root of $A(T, \mathbf{x}, z)$ of order k*

- *$A(T, \mathbf{x}, \Phi(T, \mathbf{x}))$ is not identically zero in $\mathbb{F}[T, \mathbf{x}]$.*

Let $\mathcal{S}$ be a (n, σ, μ, ρ)-design and let g be a σ-variate polynomial of degree d. For a new tuple $\mathbf{w}$ of μ variables distinct from $T, \mathbf{x}, z$, let us define the polynomial map $\mathbf{KI}_{g,\mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ using Definition 3.14. Then, the following statement is true.

If $A(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \Phi(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w})))$ is identically zero, then g can be computed by an algebraic circuit of depth $\Delta' = (\Delta + O(1))$ and size s' which is at most

$$\text{poly}(s, \deg(A), k) \cdot (\rho \log k)^{\text{poly}(d)}.$$

In particular, if g cannot be computed by a size s' depth Δ' circuit, then $A(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \Phi(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w})))$ is identically zero if and only if $A(T, \mathbf{x}, \Phi(T, \mathbf{x}))$ is identically zero.

The proof of this theorem is based on some of the same principles as that of the proof of Theorem 5.1. However, the underlying technical details are considerably shorter and cleaner. We now discuss the proof.

Proof of Theorem 7.3. Let $\hat{B}_j(T, \mathbf{x}, \mathbf{w})$ be the polynomial obtained by replacing the variables $x_1, \ldots, x_{j-1}$ in the polynomial $A(T, \mathbf{x}, \Phi(T, \mathbf{x}))$ by the polynomials $g_1, \ldots, g_{j-1} \in \mathbb{F}[\mathbf{w}]$ respectively. From the hypothesis of the theorem, we know that $\hat{B}_0$ is not identically zero and $\hat{B}_n$ is identically zero. Thus, there must be a j such that $\hat{B}_j$ is not identically zero but $\hat{B}_{j+1}$ is identically zero. We focus on this j for the rest of the proof.

Note that $\hat{B}_j$ depends on the variables $T, \mathbf{w}$ and $x_j, x_{j+1}, \ldots, x_n$ but does not depend on the variables $x_1, x_2, \ldots, x_{j-1}$. Moreover, $\hat{B}_{j+1}$ is obtained from $\hat{B}_j$ by substituting x_j by g_j. For ease of notation, we refer to the variable x_j as y for the rest of this argument. Let B_j be the polynomial obtained from $\hat{B}_j$ by setting the variables $x_{j+1}, \ldots, x_n$ and $\{w_i : w_i \notin S_j\}$ to constants from $\mathbb{F}$ such that B_j remains non-zero. Since $\hat{B}_j$ is a non-zero polynomial, a random substitution of these variables from $\mathbb{F}$ (assuming it is large enough) has this property. Thus, B_j only depends on the variables $T, y, \{w_i : w_i \in S_j\}$. For ease of notation, we continue to refer to the tuple $\mathbf{w}_{S_j}$ as $\mathbf{w}$.

We know that when we substitute y by g_j in B_j, we end up with the identically zero polynomial. In other words, $(y - g_j)$ divides B_j. At this point, we would like to invoke Lemma 7.1 to conclude the proof. In order to do that, we need to make sure that the hypothesis of the lemma holds, which we do now.

From its definition, we know that

$$\hat{B}_j(T, \mathbf{x}, \mathbf{w}) := A(T, g_1(\mathbf{w}), \ldots, g_{j-1}(\mathbf{w}), x_j, x_{j+1}, \ldots, x_n) \,.$$

Now, to obtain B_j from $\hat{B}_j$, we set the variables $x_{j+1}, \ldots, x_n$ and the $\mathbf{w}$ variables outside the set S_j to field constants, and rename x_j as y. For every $i < j$, the above setting of $\mathbf{w}$ variables outside the set S_j to field constants reduces each g_i to a polynomial $\hat{g}_i$ that has degree at most d and only depends on at most ρ $\mathbf{w}$ variables in the set $|S_i \cap S_j|$. Thus, $\hat{g}_i$ can be written as a sum of at most $\binom{\rho+d}{d} \leq \rho^d$ many monomials, and hence is a depth 2 algebraic circuit of size at most $\rho^{O(d)}$.

For $i \in [j+1, n]$, let the variable x_i be set to the field element α_i, and let $\hat{A}(T, \mathbf{w}, y)$ and $\hat{\Phi}(A, \mathbf{w}, y)$ be the polynomials defined as

$$\hat{A}(T, \mathbf{w}, y, z) := A\left(T, \hat{g}_1(\mathbf{w}), \ldots, \hat{g}_{j-1}(\mathbf{w}), y, \alpha_{j+1}, \ldots, \alpha_n, z\right) \,,$$

and

$$\hat{\Phi}(T, \mathbf{w}, y) := \Phi\left(T, \hat{g}_1(\mathbf{w}), \ldots, \hat{g}_{j-1}(\mathbf{w}), y, \alpha_{j+1}, \ldots, \alpha_n\right) \,.$$

Thus, we also get that $B_j(T, \mathbf{w}, y) = \hat{A}(T, \mathbf{w}, y, \hat{\Phi})$. A direct application of Lemma 6.1 tells us that

$\hat{A}(T, \mathbf{w}, y, z)$ is T-regularized and monic in z, and $\hat{\Phi}$ is a truncated, non-degenerate, approximate z-root of $\hat{A}(T, \mathbf{w}, y, z)$ of order k. Also, since the degree of each $\hat{g}_i$ is at most d, we have that the total degree of $\hat{A}$ is at most $d \cdot \deg(A)$. Since A has a circuit of depth Δ and size s, from the discussion above, we can plug in a small depth-2 circuit of size at most $\rho^{O(d)}$ for each $\hat{g}_i$, and get that $\hat{A}$ has a circuit of depth $\Delta + 2$ and size $\left(s\rho^{O(d)} \right)$.

So, the hypothesis of Lemma 7.1 continue to hold if we set the polynomials P and R in Lemma 7.1 to $\hat{A}$ and $\hat{\Phi}$ respectively, with size and depth bound on the circuit complexity of $\hat{A}$ being $\left(s\rho^{O(d)} \right)$ and $\Delta + 2$ respectively.

Finally, since $(y - g_j(\mathbf{w}))$ divides $B_j(T, \mathbf{w}, y)$, which equals $(\hat{A}(T, \mathbf{w}, y, \hat{\Phi}(T, \mathbf{w}, y)))$, we get from an application of Lemma 7.1 that $g_j(\mathbf{w})$ has a depth $\Delta' = (\Delta + O(1))$ circuit of size s' which is at most

$$\left(\mathrm{poly}(s\rho^d, \deg(\hat{A}), k) \cdot (\log k)^{\mathrm{poly}(d)} \right) \leq \left(\mathrm{poly}(s, \deg(A), k) \cdot (\rho \log k)^{\mathrm{poly}(d)} \right) .$$

The contrapositive tells us that if every depth-Δ' circuit for g requires size greater than s' obtained as an upper bound above, then $A(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \Phi(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w})))$ is identically zero if and only if $A(T, \mathbf{x}, \Phi(T, \mathbf{x}))$ is identically zero. This completes the proof of the theorem. □

8 Preserving irreducibility under variable reduction

We prove our main technical result (Theorem 5.1) in this section. Together with the ideas discussed in Section 9, this is sufficient to complete the proof of correctness of our algorithms. To this end, we first start by showing that the irreducibility of factors of multivariate polynomials can be characterized by certain divisibility tests involving approximate power series roots of the polynomial. We then show that these divisibility tests can be reduced to polynomial identity tests for certain circuits, where the instances are built using constant-depth circuits and their approximate power series roots. A reduction from divisibility testing to PIT was shown by Forbes [For15] and was later used in factorization algorithms in [KRS23, KRSV24]. However, our proof for this reduction here is different from that in [For15] and goes via a recent result of Andrews & Wigderson [AW24] that shows that the transformation between elementary symmetric polynomials and power symmetric polynomials can be done efficiently using constant-depth circuits. This alternative proof offers a clearer insight into the structure of the PIT instances and this structure if helpful in completing our proof. Finally, we show that these PIT instances can be solved deterministically in subexponential time using the Kabanets-Impagliazzo generator invoked with the explicit low degree hard polynomials in [LST21]. This analysis is the core technical part of the proof and is perhaps a little surprising since while we are unable to show that the PIT instances we work with are themselves constant-depth circuits. Nevertheless, we manage to show that the generator can still be analyzed and shown to work for such circuits.

8.1 From irreducibility testing to divisibility tests

Throughout this section, we assume that $P(T, \mathbf{x}, z)$ is a nonzero squarefree polynomial that is monic in z, T-regularised with respect to z, and it satisfies that $P(0, \mathbf{x}, z) = P(0, \mathbf{0}, z)$ is squarefree. Let $D = \deg_z P$ and let $P(T, \mathbf{x}, z) = \prod_{i \in [D]} (z - \varphi_i(T, \mathbf{x}))$ be the factorization[9] of $P(T, \mathbf{x}, z)$ where $\varphi_i(T, \mathbf{x})$ are from the ring of power series $\overline{\mathbb{F}}[\mathbf{x}][\![T]\!]$.

Lemma 8.1. *Let $P(T, \mathbf{x}, z)$ and $\varphi_1(T, \mathbf{x}), \dots, \varphi_D(T, \mathbf{x})$ be defined as above, and let $k > \deg_T(P)$. For any $G(T, \mathbf{x}, y) \in \overline{\mathbb{F}}(T, \mathbf{x}, z)$ that divides $P(T, \mathbf{x}, z)$, there is some subset $U \subseteq [D]$ such that*

$$G(T, \mathbf{x}, y) = \prod_{i \in U} (z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k.$$

and $\prod_{i \in U}(z - \varphi_i(T, \mathbf{x}))$ is in fact a polynomial (and not just a power series) that equals $G(T, \mathbf{x}, y)$.

Proof. Since $P(T, \mathbf{x}, z)$ is squarefree and we have the factorisation $P(T, \mathbf{x}, z) = \prod_{i \in [D]}(z - \varphi_i(T, \mathbf{x}))$ in $\mathbb{F}[\mathbf{x}, z][\![T]\!]$ and hence any polynomial factor $G(T, \mathbf{x}, z)$ must be $\prod_{i \in U}(z - \varphi_i(T, \mathbf{x}))$ for some $U \subseteq [D]$. Since $\deg_T G \leq \deg_T P < k$, we have

$$
\begin{aligned}
G(T, \mathbf{x}, y) &= G \text{ trunc } T^k \\
&= \prod_{i \in U} (z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k
\end{aligned}
$$
□

For our algorithm, we would be interested in divisibility over $\mathbb{F}$ and not over $\overline{\mathbb{F}}$. The following lemma gives some sufficient conditions for when the above polynomial $Q_S(T, \mathbf{x}, z)$ has coefficients in $\mathbb{F}$.

Lemma 8.2. *Let $P(T, \mathbf{x}, z)$ be a polynomial and $\varphi_1(T, \mathbf{x}), \dots, \varphi_D(T, \mathbf{x})$ be power series as defined above. For each $i \in [D]$, let $\alpha_i = \varphi_i(0, \mathbf{0})$, and let $\mathcal{Z} = \{\alpha_1, \dots, \alpha_D\}$ (which are the roots of $P(0, \mathbf{x}, z) = P(0, \mathbf{0}, z)$ in the algebraic closure $\overline{\mathbb{F}}$). Fix some $k \geq 0$. For a subset $S \subset [D]$, let*

$$Q_S(T, \mathbf{x}, z) := \prod_{i \in S} (z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k$$

for some $k \geq 0$.

If the polynomial $g_S(z) = \prod_{i \in S}(z - \alpha_i) = Q_S(0, \mathbf{0}, z) = Q_S(0, \mathbf{x}, z)$ has all its coefficients in the base field $\mathbb{F}$, then so does $Q_S(T, \mathbf{x}, z)$.

---

[9]Such a factorization exists because each root of the univariate $P(0, \mathbf{x}, z)$ can be extended to a power series root via Newton iteration; see [DSS22, Theorem 4].

Proof. Define the following rational functions $\Gamma^{(i)}(T, \mathbf{x}, \zeta)$ for $i \geq 0$ as follows:

$$\Gamma^{(0)}(\zeta) := \zeta$$

$$\Gamma^{(i+1)}(\zeta) := \Gamma^{(i)}(\zeta) - \frac{P(T, \mathbf{x}, \Gamma^{(i)}(\zeta))}{\partial_z P(0, \mathbf{0}, \zeta)} \quad \text{for } i > 0.$$

Note that $\Gamma^{(i)}(\alpha)$ is in fact a polynomial in $T, \mathbf{x}$ for every α such that $\partial_z P(0, \mathbf{0}, \alpha) \neq 0$. By Lemma 4.5, for any $\alpha_i \in \mathcal{Z}$, we have that $\Gamma^{(k)}(\alpha_i) = \varphi_i \bmod T^k$ and therefore

$$Q_S(T, \mathbf{x}, z) = \prod_{i \in S}(z - \Gamma^{(k)}(\alpha_i)) \text{ trunc } T^k$$

Therefore, every coefficient of $Q_S(T, \mathbf{x}, z)$ is a symmetric function of the set $\{\alpha_i : i \in S\}$. Since $g(z) = \prod_{i \in S}(z - \alpha_i)$ is assumed to have coefficients in the base field $\mathbb{F}$, this implies that every elementary symmetric function of the set $\{\alpha_i : i \in S\}$ is in $\mathbb{F}$. By the fundamental theorem of symmetric polynomials, this implies that every coefficient of $Q_S(T, \mathbf{x}, z)$ is also in $\mathbb{F}$. $\qquad \square$

With the above two lemmas, we can now characterise the factors (over $\mathbb{F}$) of $P(T, \mathbf{x}, z)$ using the polynomials $Q_S(T, \mathbf{x}, z)$ defined above.

Lemma 8.3. *Let $P(T, \mathbf{x}, z)$ and $\varphi_1(T, \mathbf{x}), \ldots, \varphi_D(T, \mathbf{x})$ be defined as above, and let $k > \deg_T P$. Let $\alpha_i = \varphi_i(0, \mathbf{0})$ for each $i \in [D]$. For a subset $S \subseteq [D]$, define the polynomial*

$$Q_S(T, \mathbf{x}, z) = \prod_{i \in S}(z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k.$$

Then, the set of factors of P over the base field $\mathbb{F}$ is the same as

$$\left\{ Q_U(T, \mathbf{x}, z) \in \overline{\mathbb{F}}[T, \mathbf{x}, z] : \begin{array}{l} U \subseteq [D] \text{ with } Q_U(T, \mathbf{x}, z) \text{ dividing } P \\ \text{and } Q_U(0, \mathbf{x}, z) = Q_U(0, \mathbf{0}, z) \in \mathbb{F}[z] \end{array} \right\}$$

Proof. Any $G(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ that is a factor of $P(T, \mathbf{x}, z)$, by Lemma 8.1, is equal to $Q_U(T, \mathbf{x}, z) = \prod_{i \in U}(z - \varphi_i(T, \mathbf{x}))$ for some $U \subseteq [D]$. Additionally, if $G(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$, then $G(0, \mathbf{x}, z) = Q_U(0, \mathbf{x}, z) \in \mathbb{F}[z]$.

For the other direction, consider any $Q_U(T, \mathbf{x}, z) \in \overline{\mathbb{F}}[T, \mathbf{x}, z]$ that divides $P(T, \mathbf{x}, z)$ and also satisfies that $Q_U(0, \mathbf{0}, z) = \prod_{i \in U}(z - \alpha_i)$ is a polynomial with coefficients in $\mathbb{F}$. By Lemma 8.2, we have that $Q_U(T, \mathbf{x}, z)$ must also be a polynomial in $\mathbb{F}[T, \mathbf{x}, z]$ and hence is a factor of $P(T, \mathbf{x}, z)$ over $\mathbb{F}$. $\qquad \square$

This immediately implies the following *certificate* for irreducibility.

Corollary 8.4. *Let $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ and $\varphi_1(T, \mathbf{x}), \ldots, \varphi_D(T, \mathbf{x})$ be defined as above, and let $k > \deg_T(P)$. Then $P(T, \mathbf{x}, z)$ is irreducible over $\mathbb{F}$ if and only if for every $\emptyset \neq S \subsetneq [D]$ such that $\prod_{i \in S}(z - \alpha_i)$*

has all coefficients in $\mathbb{F}$, *we have that* $Q_S(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ *does not divide* $P(T, \mathbf{x}, z)$ *where*

$$Q_S(T, \mathbf{x}, z) = \prod_{i \in S} (z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k.$$

Remark 8.5. *Essentially the same proof as Lemma 8.3 also characterizes the set of factors of* $P(T, \mathbf{x}, z)$ *over the algebraic closure* $\overline{\mathbb{F}}$, *and thus gives a certificate for* absolute irreducibility *using divisibility tests over* $\overline{\mathbb{F}}$. ◇

Corollary 8.6. *Let* $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ *and* $\varphi_1(T, \mathbf{x}), \dots, \varphi_D(T, \mathbf{x})$ *be defined as above, and let* $k > \deg_T(P)$. *Then* $P(T, \mathbf{x}, z)$ *is absolutely irreducible if and only if for every* $\varnothing \neq S \subsetneq [D]$, *we have that* $Q_S(T, \mathbf{x}, z) \in \overline{\mathbb{F}}[T, \mathbf{x}, z]$ *does not divide* $P(T, \mathbf{x}, z)$ *where*

$$Q_S(T, \mathbf{x}, z) = \prod_{i \in S} (z - \varphi_i(T, \mathbf{x})) \text{ trunc } T^k.$$

Corollary 8.1 together with the reduction from divisibility testing to PIT in the next section gives us a reduction from irreducibility testing to PIT. However, we note that gives exponentially many (in degree D) divisibility testing instances, and thus it isn't immediately clear if this reduction can be useful for obtaining subexponential time irreducible testing algorithms. It is for this reason that our algorithm for factorization does not proceed directly using this reduction and only uses the corollary (and the previous lemmas) in its analysis.

8.2 From divisibility testing to polynomial identity tests

Let us recall the standard definitions of the power sum symmetric polynomials and the elementary symmetric polynomials, particularly in the context of roots of a univariate polynomial.

Definition 8.7. *For any monic univariate polynomial* $P = \prod_{j \in [d]} (z - \alpha_j) \in \mathbb{F}[z]$ *and natural number* i, $\mathbf{Psym}_i(P)$ *and* $\mathbf{Esym}_i(P)$ *are the power sum symmetric polynomial* $\sum_j \alpha_j^i$ *and the elementary symmetric polynomial* $\sum_{S \subseteq [d]:|S|=i} \prod_{j \in S} \alpha_j$ *of degree* i *respectively in the multiset of roots of* P. ◇

The following important lemma from a recent beautiful work of Andrews and Wigderson [AW24] (but also present in earlier works such as [SW01]) shows that there is a constant-depth circuit that computes the power sum symmetric polynomials of a multiset from the elementary symmetric polynomials of the multiset; similarly, there is a constant-depth circuit that computes the elementary symmetric polynomials of a multiset from the power sum symmetric polynomials of the multiset.

Lemma 8.8 ([AW24, Lemma 3.4, Lemma 3.6]). *Let* $\mathbb{F}$ *be any field of characteristic zero and* $n \in \mathbb{N}$ *be any natural number. Then, there is a constant-depth circuit of size and degree* $\mathrm{poly}(n)$ *that takes the n-variate polynomials* $\{\mathbf{Esym}_i(\mathbf{x}) : i \in [n]\}$ *as inputs and outputs the polynomials* $\{\mathbf{Psym}_i(\mathbf{x}) : i \in [n]\}$.

Similarly, there is a constant-depth circuit of size and degree $\mathrm{poly}(n)$ *that takes the n-variate polynomials* $\{\mathbf{Psym}_i(\mathbf{x}) : i \in [n]\}$ *as inputs and outputs the polynomials* $\{\mathbf{Esym}_i(\mathbf{x}) : i \in [n]\}$.

In [For15, Section 7.2], Forbes showed that the question of deterministic divisibility testing for polynomials from a certain complexity class can be reduced to deterministic PIT for polynomials from a related complexity class. This reduction was extensively used in [KRS23] and [KRSV24]. The following version offers a different approach for reducing divisibility testing to PIT, using Lemma 8.8.

Lemma 8.9. *Let* $D \geq t \geq 0$ *be integer parameters. Let* $\mathbb{F}$ *be any field of characteristic zero. Then, there is a constant-depth* $\mathrm{poly}(D, t)$-*sized circuit* $\mathrm{DivTest}_{D,t}$ *on* $D + t + 1$ *variables, that takes* $(D + t)$ *inputs labelled* $f_0, \ldots, f_{D-1} \in \mathbb{F}$ *and* $g_0, \ldots, g_{t-1} \in \mathbb{F}$ *respectively, such that*

$$\mathrm{DivTest}_{D,t}(z, f_0, \ldots, f_{D-1}, g_0, \ldots, g_{t-1}) = 0$$

if and only if the polynomial $f(z) = f_0 + f_1 z + \cdots + f_{t-1} z^{t-1} + z^t$ *divides the polynomial* $g(z) = g_0 + g_1 z + \cdots + g_{D-1} z^{D-1} + z^D$.

Proof. By Lemma 8.8, we can construct multi-output circuits $\mathrm{PSymToESym}_n$ and $\mathrm{ESymToPSym}_n$ of size $\mathrm{poly}(n)$ and depth $O(1)$ such that for every choice of $\alpha_1, \ldots, \alpha_n$ we have

$$\mathrm{ESymToPSym}_n(E_1, \ldots, E_n) = (P_1, \ldots, P_n)$$
$$\mathrm{PSymToESym}_n(P_1, \ldots, P_n) = (E_1, \ldots, E_n)$$
$$\text{where } E_i = \mathbf{Esym}_i(\alpha_1, \ldots, \alpha_n)$$
$$\text{and } P_i = \mathbf{Psym}_i(\alpha_1, \ldots, \alpha_n).$$

Let $f(z) = \prod_{i=1}^{D}(z - \alpha_i)$ where $S_f = \{\alpha_1, \ldots, \alpha_D\}$ is the multiset of roots of f (in the algebraic closure of $\mathbb{F}$), and let S_g be the multiset of roots of $g(z)$.

Note that $e_i^{(f)} := \mathbf{Esym}_i(S_f) = (-1)^i f_{D-i}$ for $i = 1, \ldots, D$ and $e_i^{(g)} = \mathbf{Esym}_i(S_g) = (-1)^i g_{t-i}$ for $i = 1 \ldots, t$. Define the *pseudo-quotient* of f and g, referred to by $\tilde{h}$ as follows:

$$\tilde{h}(z) = z^{D-t} - z^{D-t-1} e_1^{(h)} + \cdots + (-1)^{D-t} e_{D-t}^{(h)}$$
$$\text{where } e_i^{(h)} = \mathrm{PSymToESym}_{D-t}(r_1, \ldots, r_{D-t})$$
$$\text{where } r_j = \left(\mathrm{ESymToPSym}_D\left(e_1^{(f)}, \ldots, e_D^{(f)}\right) - \mathrm{ESymToPSym}_t\left(e_1^{(g)}, \ldots, e_t^{(g)}\right) \right)_j$$

If $g \mid f$, then the multiset S_f is the union of the multiset S_g and the multiset S_h of roots of

$h = f/g$. In that case, for each $i = 1, \ldots, D - t$, we have

$$\mathbf{Psym}_i(S_f) = \mathbf{Psym}_i(S_g) + \mathbf{Psym}_i(S_h)$$
$$\implies \mathbf{Psym}_i(S_h) = \mathbf{Psym}_i(S_f) - \mathbf{Psym}_i(S_g).$$

Therefore, $\tilde{h}(z) = h(z)$ if $g \mid f$.

The circuit $\mathrm{DivTest}_{D,t}(f_0, \ldots, f_{D-1}, g_0, \ldots, g_{t-1})$ outputs the polynomial $f(z) - \tilde{h}(z) \cdot g(z)$ where $\tilde{h}(z)$ is computed as stated above. By construction, this is a circuit of size $\mathrm{poly}(D, t)$ and depth $O(1)$. Furthermore, as argued above, if $g \mid h$ then $\tilde{h}(z) = h(z) = f(z)/g(z)$ and hence the above computes the zero polynomial. If $g(z) \nmid f(z)$, then $f(z) - \tilde{h}(z)g(z)$ is nonzero for every choice of $\tilde{h}(z)$ and hence the above is nonzero polynomial. $\qquad\square$

The following lemma uses Lemma 8.9 to show that we can reduce the question of checking whether a candidate factor (constructed using various approximate roots from Newton iteration) is an actual factor, to a PIT instance.

Lemma 8.10. *Let $\mathbb{F}$ be a field of characteristic zero. Let $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ be a polynomial that is monic in z. Let $R_1(T, \mathbf{x}), R_2(T, \mathbf{x}), \ldots, R_\ell(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials of T-degree at most $(k-1)$.*

Then, there exists a circuit $\mathrm{DivTest}_{\deg_z(P), \ell}$ on at most $(\deg_z(P) + \ell + 1)$-variables and size and degree at most $\mathrm{poly}(\deg(P), \ell, k)$ that is computable by a depth-$(\Delta + O(1))$ such that

$$Q(T, \mathbf{x}, z) := \left(\prod_{i \in [\ell]} (z - R_i(T, \mathbf{x})) \right) \mathrm{trunc}\ T^k$$

divides $P(T, \mathbf{x}, z)$ if and only if

$$\mathrm{DivTest}_{\deg_z(P), \ell} \left(z, P_0(T, \mathbf{x}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}), (\mathbf{Esym}_1(\mathcal{R})\ \mathrm{trunc}\ T^k), \ldots, (\mathbf{Esym}_\ell(\mathcal{R})\ \mathrm{trunc}\ T^k) \right) \equiv 0,$$

where, for every i, $P_i(T, \mathbf{x})$ is the coefficient of z^i in P, when viewing it as a univariate in z with coefficients in the ring $\mathbb{F}[T, \mathbf{x}]$ and $\mathcal{R} = \{R_1, R_2, \ldots, R_\ell\}$.

Proof. This almost immediately follows from Lemma 8.9 since each of the polynomials $\mathbf{Esym}_1(\mathcal{R})\ \mathrm{trunc}\ T^k, \ldots, \mathbf{Esym}_\ell(\mathcal{R})\ \mathrm{trunc}\ T^k$ are the coefficients of $\left(\prod_{i \in [\ell]} (z - R_i(T, \mathbf{x})) \right)\ \mathrm{trunc}\ T^k$. As it is, the DivTest circuit would capture divisibility when the coefficients of the input (univariate) polynomials (in the variable z) are from a field, but since we are dealing with monic polynomials, Gauss' Lemma (Lemma 3.11) ensures that $Q(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$ in $\mathbb{F}(T, \mathbf{x})[z]$ if and only if $Q(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$ in $\mathbb{F}(T, \mathbf{x})[z]$. The theorem now follows. $\qquad\square$

Lemma 8.11. *Let $\mathbb{F}$ be a field of characteristic zero. Let $R_1(T, \mathbf{x}), R_2(T, \mathbf{x}), \ldots, R_\ell(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials of T-degree at most $(k-1)$.*

Then, there is a multi-output algebraic circuit $\hat{C}$ with depth $(\Delta + O(1))$ and size $\mathrm{poly}(\ell, k, \deg(P))$ that takes as input the polynomials of the form $\{R_i(\alpha_j T, \mathbf{x}) : i \in [\ell], j \in [\mathrm{poly}(\ell, k, \deg(P))]\}$ and outputs $\left(\mathbf{Esym}_i(R_1, R_2, \dots, R_\ell) \text{ trunc } T^k\right)$ for every $i \in [\ell]$. Here $\{\alpha_j : j \in [\mathrm{poly}(\ell, k, \deg(P))]\}$ are elements of $\mathbb{F}$.

Proof. From Lemma 8.8, we get that there is a multi-output constant-depth circuit C on ℓ variables that can be computed by a constant-depth circuit of size and degree $\mathrm{poly}(\ell)$ such that

$$C(\mathbf{Psym}_1(\mathcal{R}), \dots, \mathbf{Psym}_n(\mathcal{R})) = (\mathbf{Esym}_1(\mathcal{R}), \dots, \mathbf{Esym}_n(\mathcal{R})),$$

where, $\mathcal{R} = (R_1, R_2, \dots, R_\ell)$. For ease of notation, we just focus on the computation of one of the $\mathbf{Esym}_j(\mathcal{R})$ and consider the equality

$$\mathbf{Esym}_j(\mathcal{R}) = C(\mathbf{Psym}_1(\mathcal{R}), \dots, \mathbf{Psym}_n(\mathcal{R})).$$

The above equality immediately gives us a constant-depth circuit that takes $\mathcal{R}$ as input and outputs their elementary symmetric polynomials. However, the goal is to compute these things modulo T^k.

To recover $\mathbf{Esym}_j(\mathcal{R})$ trunc T^k from the above equality, we think of each $R_i(T, \mathbf{x})$ as a polynomial in $\mathbb{F}[\mathbf{x}][T]$ and apply Corollary 3.4 with respect to the variable T since we want to extract all the monomials that have T-degree at most k. This only incurs a polynomial blow up in size and an additive constant increase in depth. Moreover, the new circuit can be seen to be taking as inputs polynomials of the form $\{R_i(\alpha_j T, \mathbf{x}) : i \in [\ell], j \in [\mathrm{poly}(\ell, k, \deg(P))]\}$. □

The following is an immediate consequence of Lemma 8.11 and Lemma 8.10.

Corollary 8.12. *Let $\mathbb{F}$ be a field of characteristic zero. Let $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ be a polynomial that is monic in z. Let $R_1(T, \mathbf{x}), R_2(T, \mathbf{x}), \dots, R_\ell(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials of T-degree at most $(k-1)$.*

Then, there exists a circuit C on at most $(\mathrm{poly}(\ell, k, \deg(P)))$-variables and size and degree at most $\mathrm{poly}(\deg(P), \ell, k)$ that is computable by a depth $(\Delta + O(1))$ circuit such that

$$\left(\prod_{i \in [\ell]} (z - R_i(T, \mathbf{x}))\right) \text{ trunc } T^k$$

divides $P(T, \mathbf{x}, z)$ if and only if

$$C\left(z, P_0(T, \mathbf{x}), \dots, P_{\deg_z(P)}(T, \mathbf{x}), \mathcal{R}\right) \equiv 0,$$

where, for every i, $P_i(T, \mathbf{x})$ is the coefficient of z^i in P, when viewing it as a univariate in z with coefficients in the ring $\mathbb{F}[T, \mathbf{x}]$ and $\mathcal{R} = (R_i(\alpha_j T, \mathbf{x}) : i \in [\ell], j \in [\mathrm{poly}(\ell, k, \deg(P))])$, with each α_j being an element of $\mathbb{F}$.

8.3 Building up to the proof of Theorem 5.1

Theorem 8.13 is our main technical theorem building up to Theorem 5.1. It shows that for the kind of circuits that show up when testing the divisibility of candidate factors, if composing such a circuit, say C, with the KI generator – instantiated with a polynomial g and a design $\mathcal{S}$ – breaks the nonzeroness of C, then the polynomial g must be "easy".

Theorem 8.13. *Let $k > 1$ be any natural number and for every $i \in [\ell]$, let $A_i(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ and $\Phi_i(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials satisfying the following properties.*

- *Each $A_i(T, \mathbf{x}, z)$ is computable by a depth Δ circuit of size s*

- *Each $A_i(T, \mathbf{x}, z)$ is T-regularized and is monic in z.*

- *For every $i \in [\ell]$, $\Phi_i(T, \mathbf{x})$ is a truncated, non-degenerate, approximate z-root of $A_i(T, \mathbf{x}, z)$ of order k.*

Let $\mathcal{S}$ be a (n, σ, μ, ρ)-design and let g be a σ-variate polynomial of degree d. For a new tuple $\mathbf{w}$ of μ variables distinct from $T, \mathbf{x}, z$, let us define the polynomial map $\mathbf{KI}_{g,\mathcal{S}}$ using Definition 3.14. Then the following is true.

For any circuit C of depth Δ and size s such that the polynomial

$$C'(T, \mathbf{x}) := C(T, \mathbf{x}, \Phi_1, \Phi_2, \ldots, \Phi_\ell)$$

is non-zero, if $C'(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}))$ is identically zero, then g can be computed by an algebraic circuit of depth $O(\Delta)$ and size

$$\mathrm{poly}(s, \deg(C'), k) \cdot (\rho \log k)^{\mathrm{poly}(d)}.$$

Proof. The proof proceeds along the lines of the proof of Theorem 7.3. In particular, we again apply the hybrid argument to the sequence of substitutions for x_i by g_i and arrive at the point j where the substituted polynomial becomes identically zero for the first time. We again rename the variable x_j as y and set any remaining $\mathbf{x}$ variables and $\mathbf{w}$ variables outside the set S_j to field constants while preserving the non-zeroness of the polynomial before the jth substitution. We use $\hat{C}$ to denote the result of applying the sequence of substitutions to the circuit C. We use $\hat{C}'$ to denote the result of applying the sequence of substitutions to the circuit C'. Clearly, it is of the form

$$\hat{C}'(T, \mathbf{w}, y) = C(T, \hat{g}_1(T, \mathbf{w}), \ldots, \hat{g}_{j-1}(T, \mathbf{w}), y, \mathbf{b}, \hat{\Phi}_1, \ldots, \hat{\Phi}_\ell) = \hat{C}(T, \mathbf{w}, y, \hat{\Phi}_1, \ldots, \hat{\Phi}_\ell),$$

where each $\hat{\Phi}_i(T, \mathbf{w}, y)$ is a polynomial in $\mathbb{F}[T, \mathbf{w}_{S_j}, y]$ that satisfies

$$\hat{\Phi}_i(T, \mathbf{w}, y) = \Phi_i(T, \hat{g}_1(T, \mathbf{w}), \ldots, \hat{g}_{j-1}(T, \mathbf{w}), y, \mathbf{b})$$

45

for field constants $\mathbf{b}$ and each $\hat{g}_i$ is obtained from g_i by setting the $\mathbf{w}$ variables outside the set S_j to field constants according to $\mathbf{b}$ and hence depends on only $|S_i \cap S_j| \leq \rho$ variables. Thus, each $\hat{g}_i$ has a depth-2 circuit of size at most $\rho^{O(d)}$. We also know that $y - g_j(\mathbf{w})$ divides $\hat{C}'$. From this, we would like to conclude that g_j has a small constant-depth circuit.

We again follow the proof of Theorem 7.3, and reduce to the case that $(y - g_j)$ is a factor of multiplicity one. To this end, we will have to work with an appropriately high enough order derivative of $\hat{C}'$ with respect to y (depending on the multiplicity). From Corollary 3.4, we get that this derivative is in the $\mathbb{F}[y]$-span of polynomials of the form $\hat{C}'(T, \mathbf{w}, \beta_i)$ with $\beta_i \in \mathbb{F}$ and $i \in [\deg_y(\hat{C}')]$, and the y-degree of the weights in this linear combination is at most $\deg_y(\hat{C}')$. Let us denote the derivative by B. Thus, we have that

$$B(T, \mathbf{w}, y) = \sum_{i=0}^{\deg_y(\hat{C}')} \hat{C}'(T, \mathbf{w}, \beta_i) \times \Lambda_i(y),$$

for univariate polynomials $\Lambda_i(y)$ of degree at most $\deg_y(\hat{C}')$, B is non-zero and satisfies

$$B(T, \mathbf{w}, g_j(\mathbf{w})) \equiv 0,$$

and

$$\frac{\partial B}{\partial y}(T, \mathbf{w}, g_j(\mathbf{w})) \neq 0.$$

By shifting the $\mathbf{w}$ variables if needed, we can assume without loss of generality that

$$\frac{\partial B}{\partial y}(T, \mathbf{0}, g_j(\mathbf{0})) \not\equiv 0.$$

Thus, from Lemma 4.10, we get that there is a $\kappa \in \mathbb{F}$ and a polynomial Q of degree at most d on $(d+1)$ variables, that satisfies

$$g_j(\mathbf{w}) \equiv Q(h_0(\kappa, \mathbf{w}), h_1(\kappa, \mathbf{w}), \ldots, h_d(\kappa, \mathbf{w})) \mod \langle \mathbf{w} \rangle^{d+1},$$

where for every i,

$$h_i(T, \mathbf{w}) = \frac{\partial B}{\partial y^i}(T, \mathbf{w}, g_j(\mathbf{0})) - \frac{\partial B}{\partial y^i}(T, \mathbf{0}, g_j(\mathbf{0})) \text{ trunc } \langle \mathbf{w} \rangle^{d+1}.$$

Given the degree of Q and the number of variables, we get that it has a depth-2 circuit of size at most $\exp(O(d))$. The next claim shows that each h_i has a constant-depth circuit of small size.

Claim 8.14. *Each h_i can be computed by a circuit of depth $O(\Delta)$ and size*

$$s' \leq \mathrm{poly}(s, \deg(C'), k) \cdot (\rho \log k)^{\mathrm{poly}(d)}$$

We first use the claim to complete the proof of the lemma, and discuss the proof of the claim.

We take the circuits for $h_i(T, \mathbf{w})$ — given by Claim 8.14, denoted by $V_i(T, \mathbf{w})$ — and consider the circuit $Q(V_0(\kappa, \mathbf{w}), V_1(\kappa, \mathbf{w}), \dots, V_d(\kappa, \mathbf{w}))$. Clearly, it is of depth $O(\Delta)$ and size $s' \leq \mathrm{poly}(s, \deg(C'), k) \cdot (\rho \log k)^{\mathrm{poly}(d)}$ and satisfies

$$g_j(\mathbf{w}) \equiv Q(V_0(\kappa, \mathbf{w}), V_1(\kappa, \mathbf{w}), \dots, V_d(\kappa, \mathbf{w})) \mod \langle \mathbf{w} \rangle^{d+1}.$$

We can now apply Corollary 3.4 to obtain the constant-depth circuit for $g_j(\mathbf{w})$ with only an additive increase in depth and a polynomial blow up in the size, thereby giving us the theorem. □

Proof of Claim 8.14. From Lemma 6.1, we know that for any $\beta \in \mathbb{F}$, the polynomials $\hat{A}_i(T, \mathbf{w}, \beta)$ and $\hat{\Phi}_i(T, \mathbf{w}, \beta)$, obtained from A and Φ using the substitutions in the above discussion and setting $y = \beta$, continue to satisfy that properties satisfied by A_i, Φ_i in the hypothesis of the lemma.

Thus, for any field constant β, we can invoke Lemma 6.3 for each $\hat{A}_i(T, \mathbf{w}, \beta)$ and $\hat{\Phi}_i(T, \mathbf{w}, \beta)$ to get that $(\hat{\Phi}_i(T, \mathbf{w}, \beta) \text{ trunc } \langle \mathbf{w} \rangle^{d+1})$ can be computed by a circuit of depth $(\Delta + O(1))$ and size s'. We now plug these circuits into the input gates of the constant-depth circuit $\hat{C}$ to get a circuit of depth at most $O(\Delta)$ and size $O(s')$ that computes the polynomial

$$\hat{C}(T, \mathbf{w}, \beta, (\hat{\Phi}_1(T, \mathbf{w}, \beta) \text{ trunc } \langle \mathbf{w} \rangle^{d+1}), \dots, (\hat{\Phi}_\ell(T, \mathbf{w}, \beta) \text{ trunc } \langle \mathbf{w} \rangle^{d+1})).$$

Now, applying Corollary 3.4 with respect to the $\mathbf{w}$ variables gives us a circuit of depth $O(\Delta)$ and size $\mathrm{poly}(s')$ that computes $(\hat{C}'(T, \mathbf{w}, \beta) \text{ trunc } \langle \mathbf{w} \rangle^{d+1})$, since by definition $\hat{C}'(T, \mathbf{w}, y)$ equals $\hat{C}(T, \mathbf{w}, y, \hat{\Phi}_1, \dots, \hat{\Phi}_\ell)$.

Similar to Lemma 7.1[10], the claim now follows from the definition of B, Corollary 3.4 and the fact that the degree of B in T is at most $\deg(C')$. □

We now combine the machinery in Section 8.2 with Theorem 8.13 to get the following theorem that is directly used in the proof of Theorem 5.1.

Theorem 8.15. *Let $n \in \mathbb{N}$ be sufficiently large, and let $\mathbf{x} = (x_1, \dots, x_n)$. Let $\mathbb{F}$ be a field of characteristic zero. Let $k > 1$ be any natural number. Let $P(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ and $R_1(T, \mathbf{x}), R_2(T, \mathbf{x}), \dots, R_\ell(T, \mathbf{x}) \in \mathbb{F}[T, \mathbf{x}]$ be polynomials with the following properties.*

- *$P(T, \mathbf{x}, z)$ is computable by a depth Δ circuit of size $s \leq \mathrm{poly}(n)$*

- *$P(T, \mathbf{x}, z)$ is T-regularized and is monic in z.*

---

[10]Refer to the part of the proof where we show that $h_i(T, \mathbf{w})$ can be computed by constant-depth circuits.

- *For every $i \in [\ell]$, $R_i(T, \mathbf{x})$ is a truncated, non-degenerate, approximate z-root of $P(T, \mathbf{x}, z)$ of order k.*

Let $Q(T, \mathbf{x}, z) := \left(\prod_{i \in [\ell]} (z - R_i(T, \mathbf{x})) \right)$ trunc T^k.

Let g be a σ-variate degree d polynomial and $\mathcal{S}$ be a (n, σ, μ, ρ)-design. Let $\mathbf{KI}_{g,\mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ be the polynomial map in Definition 3.14 defined using g and $\mathcal{S}$. Then, the following is true.

If $Q(T, \mathbf{x}, z)$ does not divide $P(T, \mathbf{x}, z)$ but $Q(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ divides $P(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$, then g can be computed by an algebraic circuit of depth $\Delta' = O(\Delta)$ and size at most

$$s' = \mathrm{poly}(s, \ell, k, \deg(P)) \cdot (\rho \log k)^{\mathrm{poly}(d)}.$$

In particular, if any depth-Δ' circuit for g requires size greater than s', then $Q(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$ if and only if $Q(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ divides $P(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$

Proof. If $Q(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$, then $Q(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ certainly divides $P(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$.

Suppose $Q(T, \mathbf{x}, z)$ does not divide $P(T, \mathbf{x}, z)$. Corollary 8.12 tells us that equivalently, there exists a circuit C on at most $\mathrm{poly}(\ell, k, \deg(P))$ variables, and size and degree at most $\mathrm{poly}(\ell, k, \deg(P))$, that is computable by a depth $(\Delta + O(1))$ circuit such that

$$C \left(z, P_0(T, \mathbf{x}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}), \mathcal{R} \right) \not\equiv 0,$$

where, for every i, $P_i(T, \mathbf{x})$ is the coefficient of z^i in P when viewing it as a univariate in z with coefficients in the ring $\mathbb{F}[T, \mathbf{x}]$, and $\mathcal{R} = \left(R_i(\alpha_j T, \mathbf{x}) : i \in [\ell], j \in [\mathrm{poly}(s, k, \deg(P))] \right)$, with each α_j being an element of $\mathbb{F}$.

If $Q(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ divides $P(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$, then

$$C \left(z, P_0(T, \mathbf{x}) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \mathcal{R} \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}) \right) \equiv 0,$$

where $\mathcal{R} \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}) := \{ R \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}) : R \in \mathcal{R} \}$. Observe that

$$C \left(z, P_0(T, \mathbf{x}) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), \mathcal{R} \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}) \right)$$

is the same as

$$C \left(z, P_0(T, \mathbf{x}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}), \mathcal{R} \right) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}).$$

Moreover, Lemma 6.1 tells us that each $R_i(\alpha_j T, \mathbf{x})$ is a truncated, non-degenerate, approximate z-root of $P(\alpha_j T, \mathbf{x}, z)$ of order k, where $P(\alpha_j T, \mathbf{x}, z)$ is T-regularized and monic in z.

Thus, we can apply Theorem 8.13 to $C \left(z, P_0(T, \mathbf{x}), \ldots, P_{\deg_z(P)}(T, \mathbf{x}), \mathcal{R} \right)$ to conclude that g can be computed by an algebraic circuit of depth $\Delta' = O(\Delta)$ and size s' which is at most

$$s' := \mathrm{poly}(s, \ell, k, \deg(P)) \cdot (\rho \log k)^{\mathrm{poly}(d)}.$$

The contrapositive tells us that if every depth-Δ' circuit for g requires size greater than s' obtained as an upper bound above, then $Q(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$ if and only if $Q(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$ divides $P(T, \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}), z)$. $\square$

8.4 Proof of Theorem 5.1

We are now ready to prove Theorem 5.1. We start by recalling some notation that we use.

Let $\mathcal{G} = \{g_m\}_{m \in \mathbb{N}}$ be a family of polynomials such that for every $m \in \mathbb{N}$, $g_m \in \mathbb{F}[x_1, \ldots, x_m]$, $d_m := \deg(g_m) \leq O(\log \log(m))$. Further, $\mathcal{G}$ has the property that for any depth $\Delta \in \mathbb{N}$, if $\mathcal{C} = \{C_m\}_{m \in \mathbb{N}}$ is a family of depth-Δ circuits computing $\mathcal{G}$, then $\mathcal{C}$ requires size $m^{d_m^{\exp(-O(\Delta))}}$, which is $m^{\omega(1)}$. Theorem 3.15 gives us such a family of explicit low-degree polynomials that are hard for constant-depth circuits.

Theorem 5.1 (Irreducibility-preserving variable reduction). *Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. For an absolute constant[11] $C_{\Delta, \varepsilon} \in \mathbb{N}$, let $n \in \mathbb{N}, n \geq C_{\Delta, \varepsilon}$ and $\mathbf{x} := (x_1, \ldots, x_n)$. Let $P(T, \mathbf{x}, z)$ be a nonzero polynomial with the following properties.*

- *$P(T, \mathbf{x}, z)$ is computable by a size $s \leq \mathrm{poly}(n)$ and depth Δ circuit.*

- *$P(T, \mathbf{x}, z)$ is monic in z and T-regularized, with $\deg(P) = D \leq \mathrm{poly}(n)$.*

- *$P(T, \mathbf{x}, z)$ and $P(0, \mathbf{x}, z) = P(0, \mathbf{0}, z)$ are squarefree.*

Let $\sigma = O(n^\varepsilon), \mu = O(\frac{n^{2\varepsilon}}{\log(n)}), \rho = O(\log(n))$, and let $\mathcal{S}$ be an (n, σ, μ, ρ)-design. Let $\mathbf{KI}_{g_\sigma, \mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ be the polynomial map in Definition 3.14 defined using the design $\mathcal{S}$ and the polynomial g_σ from the family of hard polynomials $\mathcal{G}$. Then, the following is true.

A polynomial $F(T, \mathbf{x}, z)$ is an irreducible factor of $P(T, \mathbf{x}, z)$ if and only if $F(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$ is an irreducible factor of $P(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$.

Proof. Suppose $P(T, \mathbf{x}, z) = \prod_{i=1}^{D}(z - \varphi_i(T, \mathbf{x}))$, where each $\varphi_i(T, \mathbf{x})$ is a power series root from $\mathbb{F}[\mathbf{x}][\![T]\!]$. Then, by Corollary 8.1, we have that for any subset $S \subseteq [D]$, $F_S(T, \mathbf{x}, z) := \prod_{i \in S}(z - \varphi_i(T, \mathbf{x}))$ is an irreducible factor of $P(T, \mathbf{x}, z)$ in $\mathbb{F}[T, \mathbf{x}, z]$ if and only if

- $Q_S(T, \mathbf{x}, z)$ divides $P(T, \mathbf{x}, z)$, where $Q_S = F_S$ trunc T^k and

- for every $\varnothing \neq U \subsetneq S$ such that $\prod_{i \in U}(z - \alpha_i)$ has all coefficients in $\mathbb{F}$, we have that $Q_U(T, \mathbf{x}, z) \in \mathbb{F}[T, \mathbf{x}, z]$ does not divide $P(T, \mathbf{x}, z)$

---

[11]If $a(n) = O(b(n))$, then there exists some C such that for all $n > C$, $a(n) \leq b(n)$; the $C_{\Delta, \varepsilon}$ in our statement is for this purpose. The precise value of $C_{\Delta, \varepsilon}$ depends on the exact hardness of the polynomial in $\mathcal{G}$ and the upper bounds obtained in our proofs.

For the given parameters, one can verify that the upper bound s' given in Theorem 8.15 is at most $\text{poly}(n)$, whereas g_σ requires size $n^{\omega(1)}$ for any constant-depth circuit[12]. Thus, we can apply Theorem 8.15 on both the conditions above, to equivalently state that for any subset $S \subseteq [D]$, $F_S(T, \mathbf{x}, z) = \prod_{i \in S}(z - \varphi_i(T, \mathbf{x}))$ is an irreducible polynomial factor of $P(T, \mathbf{x}, z)$ if and only if

- $Q_S(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$ divides $P(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$, and

- for every $\emptyset \neq U \subsetneq S$ such that $\prod_{i \in U}(z - \alpha_i)$ has all coefficients in $\mathbb{F}$, we have that $Q_U(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$ does not divide $P(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$

Lemma 6.1 tells us that $P(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$ will remain a nonzero squarefree[13] polynomial that is monic in z and T-regularized, with $\deg_z P = D$. Furthermore, $P(0, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z) = P(0, \mathbf{0}, z)$ will be squarefree. Thus, we can apply Corollary 8.1 to the above two conditions to get that for any subset $S \subseteq [D]$, $F_S(T, \mathbf{x}, z) = \prod_{i \in S}(z - \varphi_i(T, \mathbf{x}))$ is an irreducible polynomial factor of $P(T, \mathbf{x}, z)$ if and only if

- $\prod_{i \in S}(z - \varphi_i(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}))) \in Q[T, \mathbf{w}, z]$, and

- for every $\emptyset \neq U \subsetneq S$ such that $\prod_{i \in U}(z - \alpha_i)$ has all coefficients in $\mathbb{F}$, we have that $\prod_{i \in T}(z - \varphi_i(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}))) \notin Q[T, \mathbf{w}, z]$

These conditions are true if and only if $F_S(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z) = \prod_{i \in S}(z - \varphi_i(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w})))$ is an irreducible factor of $P(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$; this concludes the theorem. $\square$

9   Proof of correctness of the algorithm

We will analyze the correctness of our algorithms in a bottom-up fashion, starting with Algorithm 3 and ending with Algorithm 1, which will prove Theorem 5.2.

9.1   Analysis of Algorithm 3

Suppose $P(T, \mathbf{x}, z) = \prod_{i \in [m]} G_i(T, \mathbf{x}, z)$, where $P(T, \mathbf{x}, z)$ is monic in z, T-regularized with $\deg_T(P) \leq \deg(P) := D$ and $\deg_z(P) = D_z$. P is squarefree and $P(0, \mathbf{0}, z)$ is squarefree as well. The G_is are the irreducible factors of P.

Let $G(T, \mathbf{x}, z)$ denote an arbitrary G_i. Suppose $G(0, \mathbf{0}, z) = \prod_{j \in [r]} H_j(z)$ is the factorization of $G(0, \mathbf{0}, z)$ into its irreducible factors. Since $P(0, \mathbf{0}, z)$ is squarefree, each H_j is distinct. If any of the $H_j(z)$ had degree 1, we could've lifted that root via Newton iteration and proceeded with the rest of the algorithm. We will now describe how we deal with the absence of roots of $G(0, \mathbf{0}, z)$ in $\mathbb{Q}$ using ideas that are standard in the literature (for instance, see [DSS22, Section 6.2]).

---

[12]We chose n sufficiently large enough in the theorem statement ($n > C$), which implies that for this particular σ, g_σ requires constant-depth circuits larger than s'.

[13]Squarefreeness is equivalent to each root being non-degenerate, which is maintained by Lemma 6.1.

Let $H(z)$ be an arbitrary irreducible factor of $G(0, \mathbf{0}, z)$ and define the field $\mathbb{K} := \frac{\mathbb{Q}[u]}{H(u)}$ for a new variable u. Thus, we are artificially adding a root of $H(u)$ to our field. Every operation in this field is polynomial addition or multiplication over $\mathbb{Q}$ with the variable u, followed by taking the reminder $\mod H(u)$.

We can now use Newton iteration (Lemma 4.5) to compute a truncated, non-degenerate, approximate z-root $\varphi(T, \mathbf{x}) \in \mathbb{K}[T, \mathbf{x}]$ satisfying $G(T, \mathbf{x}, \varphi(T, \mathbf{x})) \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$ and $\varphi(0, \mathbf{x}) = u \in \mathbb{K}$. The following lemma, standard in the factorization literature (see [Bü04, Lemma 3.6]), tells us that G is the unique minimal polynomial of φ with the above constraints.

Lemma 9.1. *Let $G(T, \mathbf{x}, z) \in \mathbb{Q}[T, \mathbf{x}, z]$ be an irreducible polynomial, monic in z with T-degree at most D and z-degree exactly D_z. Let $\varphi(T, x) \in \mathbb{K}[T, \mathbf{x}]$ be an approximate root of G of order $2D \cdot D_z + 1$, satisfying $G(T, \mathbf{x}, \varphi(T, \mathbf{x})) \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$.*

Now, suppose $B(T, z) \in \mathbb{Q}(\mathbf{x})[T, z]$ is monic in z with T-degree at most D and z-degree exactly D_z satisfying:

$$B(T, \varphi(T, \mathbf{x})) \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$$

Then $B \equiv G$.

Proof. Consider $R(T) := \text{Res}_z(B(T, z), G(T, z)) \in \mathbb{Q}(\mathbf{x})[T]$. By Theorem 3.8, there exist polynomials $A_B(T, z), A_G(T, z) \in \mathbb{Q}(\mathbf{x})[T, z]$ such that

$$R(T) \equiv A_B(T, z)B(T, z) + A_G(T, z)G(T, z).$$

If we plug in $z = \varphi$, both $B(T, \varphi)$ and $G(T, \varphi)$ vanish mod $\langle T \rangle^{2D \cdot D_z + 1}$. Thus, $R(T) \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$. But the definition of the Resultant tells us that $R(T)$ has T-degree at most $2D \cdot D_z$, which means that $R(T)$ must be identically zero. By Theorem 3.8, this implies that $\gcd(B, G)$ is non-trivial in $\mathbb{Q}(T, \mathbf{x})[z]$, but since both B and G are monic in z, Lemma 3.11 tells us that $\gcd(B, G)$ is non-trivial in $\mathbb{Q}[T, \mathbf{x}, z]$. Since G is irreducible, G must divide B. Moreover, $\deg_z(B) = \deg_z(G)$, which implies that $B \equiv G$. $\square$

9.1.1 Linear system for computing the unique minimal polynomial

Let $\mathcal{G} = \{g_m\}_{m \in \mathbb{N}}$ be a family of polynomials such that for every $m \in \mathbb{N}$, $g_m \in \mathbb{F}[x_1, \ldots, x_m]$, $d_m := \deg(g_m) \leq O(\log \log(m))$. Further, $\mathcal{G}$ has the property that for any depth $\Delta \in \mathbb{N}$, if $\mathcal{C} = \{C_m\}_{m \in \mathbb{N}}$ is a family of depth-Δ circuits computing $\mathcal{G}$, then $\mathcal{C}$ requires size $m^{d_m^{\exp(-O(\Delta))}}$, which is $m^{\omega(1)}$. Theorem 3.15 gives us such a family of explicit low-degree polynomials that are hard for constant-depth circuits.

Theorem 9.2 (Small division-free circuit for the minimal polynomial of an approximate root). *Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. For an absolute constant $C_{\Delta,\varepsilon} \in \mathbb{N}$, let $n > C_{\Delta,\varepsilon}$ and $\mathbf{x} = (x_1, \ldots, x_n)$. Let $P(T, \mathbf{x}, z)$ be a polynomial with the following properties.*

- *P is T-regularized and monic in z with total degree D.*

- *$G(T, \mathbf{x}, z)$ is an irreducible factor of P with z-degree D_z.*

- *$H(z)$ is an irreducible factor of $G(0, \mathbf{x}, z)$, and $\mathbb{K}$ is the field $\frac{\mathbb{Q}[u]}{H(u)}$.*

Then, there is a deterministic algorithm $\mathcal{A}_{\Delta,\varepsilon}$ (with access to $H(z)$) which

- *takes as input a size s circuit computing $\varphi(T, \mathbf{x}) \in \mathbb{K}[T, \mathbf{x}]$, a truncated, approximate z-root of $P(T, \mathbf{x}, z)$ of order $(2D \cdot D_z + 1)$ such that $\varphi(0, \mathbf{x}) = u$ is a root of $G(0, \mathbf{x}, z)$ over $\mathbb{K}$;*

- *outputs a division-free circuit over $\mathbb{Q}$ for $G(T, \mathbf{x}, z)$ with size $\mathrm{poly}(s, D)$;*

- *and runs in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$.*

Proof. We would like to compute a polynomial $B(T, z) \in \mathbb{Q}(\mathbf{x})[T, z]$ that is monic in z with T-degree at most D and z-degree exactly D_z satisfying:

$$B(T, \varphi(T, \mathbf{x})) \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$$

where $\varphi(T, \mathbf{x}) \in \mathbb{K}[T, \mathbf{x}]$ is computed by a circuit with constants from $\mathbb{K}$. Lemma 9.1 guarantees that such a polynomial B has to be the irreducible factor G.

We can separate B into its bihomogeneous components with respect to T and z as

$$B(T, z) = \sum_{i=0}^{d} \sum_{j=0}^{D} B_{i,j} z^i T^j$$

where each $B_{i,j}$ can take a value in the field $\mathbb{Q}(\mathbf{x})$. Since B is monic in z, $B_{d,0} = 1$ and $B_{d,j} = 0$ for all $j > 0$. The minimal polynomial satisfies

$$B(T, \varphi(T, \mathbf{x})) = \sum_{i=0}^{d} \sum_{j=0}^{D} B_{i,j} (\varphi(T, \mathbf{x}))^i T^j \equiv 0 \mod \langle T \rangle^{2D \cdot D_z + 1}$$

which can equivalently be written as $2D \cdot D_z + 1$ many linear constraints, where each constraint expresses that the coefficient of T^l in $B(T, \varphi(T, \mathbf{x}))$ is zero, for some $l \in \{0, 1, \ldots, 2D \cdot D_z\}$. Let $\varphi^{(i,j)}$ denote the coefficient of T^j in φ^i. Thus, we have the following linear system in the variables $B_{i,j}$ for

$i \in \{0, \ldots, D\}$ and $j \in \{0, \ldots, D_z\}$.

$$B_{d,0} = 1 \tag{9.3}$$

$$\forall j \in \{1, \ldots, D\} : B_{d,j} = 0 \tag{9.4}$$

$$\forall l \in \{0, 1, \ldots, 2D \cdot D_z + 1\} : \sum_{i=0}^{d} \sum_{j=0}^{l} B_{i,j}(\varphi)^{(i,l-j)} = 0 \tag{9.5}$$

At this point, the coefficients of each constraint come from $\mathbb{K}[\mathbf{x}]$, which means that we can interpret each constraint as a degree $(\deg(H) - 1)$ polynomial in the variable u, with coefficients from $\mathbb{Q}[\mathbf{x}]$. Since the minimal polynomial of u is H, it follows that $1, u, \ldots, u^{\deg(H)-1}$ are linearly independent over $\mathbb{Q}$ and, in fact, over[14] $\mathbb{Q}[\mathbf{x}]$. Thus, for a constraint to be equal to zero, the coefficient of each u^r (for $r \in \{0, 1, \ldots, \deg(H) - 1\}$) must be identically zero in $\mathbb{Q}[\mathbf{x}]$. To compute the coefficient of u^r in $(\varphi)^{(i,j)}$, we can apply Corollary 3.4 and get a circuit of size $\text{poly}(s)$; we will denote this circuit by $(\varphi)^{(i,j,r)}$. Thus, we can express the constraints of the form Eq. (9.5) using equivalent constraints of the form:

$$\sum_{i=0}^{d} \sum_{j=0}^{l} B_{i,j}(\varphi)^{(i,l-j,r)} = 0$$

for every $r \in \{0, \ldots, (\deg(H) - 1)D_z\}$ and for every $l \in \{0, 1, \ldots, 2D \cdot D_z + 1\}$. The linear system is now over $\mathbb{Q}[\mathbf{x}]$, and thus, any solution to the linear system is going to be over $\mathbb{Q}(\mathbf{x})$.

We can express the linear system as $M_\varphi B = c_\varphi$ for a matrix M_φ with entries in $\mathbb{Q}[\mathbf{x}]$, the vector of variables $B = (B_{i,j})$ and a vector $c_\varphi \in (\mathbb{Q}[\mathbf{x}])^n$. Since the linear system has a unique solution (Lemma 9.1), M_φ has full column rank. Using some basic linear algebra (for instance, see [KRSV24, Lemma B.6]), we can rewrite the linear system as $M_\varphi^T M_\varphi B = M_\varphi^T c_\varphi$, where $M_\varphi^T M_\varphi$ is an invertible square matrix. Thus, we can express the solution to this linear system as $B = (M_\varphi^T M_\varphi)^{-1}(M_\varphi^T c_\varphi)$. In particular, each $B_{i,j}$ can be expressed as $\frac{N_{i,j}(\mathbf{x})}{\det(M_\varphi^T M_\varphi)}$, where both the numerator and the denominator have circuits of size $\text{poly}(s, D)$.[15]

At this point, we would like to use Strassen's division elimination (Lemma 3.6) to write each $B_{i,j}$ as a circuit without division. Strassen's method requires that we find a point $\gamma \in \mathbb{Q}^n$ such that $\det(M_\varphi^T M_\varphi)$ is non-zero at γ, but this seems to require a hitting set for circuits since the naive upper bound that we can give for $\det(M_\varphi^T M_\varphi)$ is a small circuit. We will now show that the variable reduction map from Theorem 5.1 also preserves the nonzeroness of $\det(M_\varphi^T M_\varphi)$.

Let $\sigma = O(n^\varepsilon), \mu = O(\frac{n^{2\varepsilon}}{\log(n)}), \rho = O(\log(n))$, and let $\mathcal{S}$ be an (n, σ, μ, ρ)-design. Let $\mathbf{KI}_{g_\sigma, \mathcal{S}} : \mathbb{F}^\mu \to \mathbb{F}^n$ be the polynomial map in Definition 3.14 defined using the design $\mathcal{S}$ and the polynomial g_σ from the family of hard polynomials $\mathcal{G}$.

---

[14]If $\sum_i p_i(\mathbf{x})u^i \equiv 0$, then $\sum_i p_i(\mathbf{0})u^i = 0$.

[15]Here, we use the well-known fact that the Determinant can be computed efficiently.

Claim 9.6. $\det(M_\varphi^T M_\varphi) \circ \mathbf{KI}_{g,\mathcal{S}}(\mathbf{w}) \neq 0$

Proof. Recall that $\det(M_\varphi^T M_\varphi)$ is non-zero precisely because the irreducible factor $G(T, \mathbf{x}, z)$ is the unique solution for the linear system. Lemma 6.1 tells us that $\varphi(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}))$ is the unique truncated, non-degenerate, approximate z-root of $P(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$ of order $2D \cdot D_z + 1$, satisfying $\varphi(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})) = u = \varphi(0, \mathbf{x})$. By Theorem 5.1, we now know that $\mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})$ preserves the irreducibility of factors, so $G(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}), z)$ is irreducible. Thus, if we had first applied $\mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})$ to the polynomial $P(T, \mathbf{x}, z)$ and computed the truncated, non-degenerate, approximate root $\varphi(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}))$ of order $2D \cdot D_z + 1$ starting from the same point $u \in \mathbb{K}$, Lemma 9.1 tells us that we would've still maintained uniqueness of solution for the linear system that computes a monic minimal polynomial of $\varphi(T, \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}))$ of T-degree at most D and z-degree exactly D_z. Moreover, the new linear system is going to be the same as the old linear system, except each $(\varphi)^{(i, l-j, r)}$ will be replaced by $(\varphi \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}))^{(i, l-j, r)}$; this follows because the transformation $\mathbf{x} \mapsto \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})$ preserves the (T, z)-degree and the u-degree of every monomial; in other words, the matrix $M_{\varphi \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})}$ will be equal to $M_\varphi \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})$. Thus, uniqueness of the linear system's solution implies that $\det(M_\varphi^T M_\varphi)(\mathbf{x}) \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w}) \equiv \det(M_{\varphi \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})}^T M_{\varphi \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})}) \neq 0$. □

Thus, to find a point $\gamma \in \mathbb{Q}^n$ satisfying $\det(M_\varphi^T M_\varphi)(\gamma) \neq 0$, we can first compose it with $\mathbf{KI}_{g_\sigma, \mathcal{S}}(\mathbf{w})$ to get a degree $\mathrm{poly}(n)$ polynomial on $\mu = O(n^{2\varepsilon})$ variables. For this low-variate polynomial, we can use the brute force derandomization of the Polynomial Identity Lemma (Lemma 3.2) to find a point $\hat{\gamma} \in \mathbb{Q}^\mu$ such that $\det(M_\varphi^T M_\varphi) \circ \mathbf{KI}_{g_\sigma, \mathcal{S}}$ is nonzero at $\hat{\gamma}$; this takes time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$. Thus, $\gamma := \mathbf{KI}_{g_\sigma, \mathcal{S}}(\hat{\gamma}) \in \mathbb{Q}^n$ will be a point where $\det(M_\varphi^T M_\varphi)(\gamma) \neq 0$. We can then use Lemma 3.6 to give a circuit of size $\mathrm{poly}(s, D)$ for each $B_{i,j}$, which implies that $G(T, \mathbf{x}, z) = B(T, z) = \sum_{i=0}^d \sum_{j=0}^D B_{i,j} z^i T^j$ has a circuit of size $\mathrm{poly}(s, D)$.

□

Theorem 9.7 (Correctness of Algorithm 3). *For any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$, Algorithm 3, for sufficiently large n, satisfies the following.*

- *It takes as input a depth-Δ size-s circuit $C_{\tilde{P}}$ computing a squarefree degree-D polynomial $\tilde{P}(T, \mathbf{x}, z) = \prod_{j \in [m]} \tilde{G}_j(T, \mathbf{x}, z)$, where the $\tilde{G}_i$s are the irreducible factors of $\tilde{P}$. Further, $\tilde{P}(T, \mathbf{x}, z)$ is monic in z, T-regularized with respect to $\mathbf{x}$ and $C_{\tilde{P}}(0, \mathbf{x}, z) = C_{\tilde{P}}(0, \mathbf{0}, z)$ is squarefree.*

- *It also takes as input the univariate polynomial $\tilde{G}_j(0, \mathbf{x}, z) = \tilde{G}_j(0, \mathbf{0}, z)$ for each $j \in [m]$.*

- *It outputs a list $L = \{C_{\tilde{G}_1}(T, \mathbf{x}, z), \ldots, C_{\tilde{G}_m}(T, \mathbf{x}, z)\}$, such that each $C_{\tilde{G}_i}(T, \mathbf{x}, z)$ is a circuit of size $\mathrm{poly}(s, D)$ and depth $\mathrm{poly}(D)$, which computes the irreducible factor $\tilde{G}_i(T, \mathbf{x}, z)$ of $\tilde{P}(T, \mathbf{x}, z)$.*

- *It runs in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$.*

Proof. The algorithm iterates over each $\tilde{G}_j$, so let us focus on one such $\tilde{G}_j$. By Theorem 4.11, Line 3 of the algorithm gives the correct factorization of $\tilde{G}_j(0, \mathbf{0}, z)$ into its irreducible factors, and it runs in time $\mathrm{poly}(\deg(\tilde{G}_j(0, \mathbf{0}, z)))$. By Corollary 4.6, Line 5 will correctly compute a truncated, non-degenerate, approximate z-root $\varphi(T, \mathbf{x}) \in \mathbb{K}[T, \mathbf{x}]$ of $\tilde{G}_j(T, \mathbf{x}, z)$ of order $2D \cdot D_z + 1$ such that $\varphi(0, \mathbf{x}) = \varphi(0, \mathbf{0}) = u \in \mathbb{K}$. Moreover, it runs in time $\mathrm{poly}(s, D)$ and outputs a circuit for $\varphi(T, \mathbf{x})$ of size $\mathrm{poly}(s, D)$. Finally, by Theorem 9.2, Line 6 will correctly compute a circuit $C_{\tilde{G}_j}$ for $\tilde{G}_j$, of size $\mathrm{poly}(s, D)$, in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$. □

9.2 Analysis of Algorithm 2

Theorem 9.8 (Correctness of Algorithm 2). *For any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$, Algorithm 2, for sufficiently large n, satisfies the following.*

- *The input to the algorithm is a depth-Δ size-s circuit C_P computing the squarefree polynomial $P(\mathbf{x}) = \prod_{j \in [m]} G_j(\mathbf{x})$, where the G_js are the irreducible factors of P.*

- *The output of the algorithm is a list $L = \{C_{G_1}(\mathbf{x}), \ldots, C_{G_m}(\mathbf{x})\}$, such that each $C_{G_j}(\mathbf{x})$ is a circuit of size $\mathrm{poly}(s, D)$ and depth $\mathrm{poly}(D)$, which computes the irreducible factor $G_j(\mathbf{x})$.*

- *The algorithm runs in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$.*

Proof. The coefficient of z^D in $\hat{P}(\mathbf{x}, z) := P(\mathbf{x} + (\mathbf{a} \cdot z))/\delta$ is $\mathrm{Hom}_D[P]/\delta = 1$, thus $\hat{P}$ is monic in z, and $\hat{P}$ retains the squarefreeness of P. Moreover, Corollary 3.4 tells us that $\mathrm{Hom}_D[P]$ has a size $\mathrm{poly}(s, D)$ and depth $(\Delta + O(1))$ circuit, so Theorem 3.16 outputs $\mathbf{a}$ satisfying $\mathrm{Hom}_D[P](\mathbf{a}) \neq 0$, in time $\mathrm{poly}(s, D)^{O(n^\varepsilon)}$.

Since $\hat{P}$ is monic in z and squarefree, Theorem 3.9 along with Lemma 3.11 tells us that $\mathrm{Disc}_z(\hat{P})(\mathbf{x}) \not\equiv 0$. In Line 2, $\mathbf{b}$ is chosen to ensure that $\mathrm{Disc}_z(\hat{P})(\mathbf{b}) \neq 0$. For $\tilde{P}(T, \mathbf{x}, z) := \hat{P}((T \cdot \mathbf{x}) + \mathbf{b}, z)$, $\mathrm{Disc}_z(\tilde{P}(0, \mathbf{x}, z)) = \mathrm{Disc}_z(\tilde{P}(0, \mathbf{0}, z)) = \mathrm{Disc}_z(\hat{P})(\mathbf{b}) \neq 0$, which implies that $\tilde{P}(0, \mathbf{x}, z) \in \mathbb{Q}[z]$ is squarefree. Moreover, Theorem 3.10 (along with an application of Corollary 3.4 for the complexity of $\frac{\partial \hat{P}}{\partial z}$) tells us that $\mathrm{Disc}_z(\hat{P}) = \mathrm{Res}_z(\hat{P}, \frac{\partial \hat{P}}{\partial z})$ has a size $\mathrm{poly}(s, D)$ and depth $(\Delta + O(1))$ circuit. Thus, we can again use Theorem 3.16 to find $\mathbf{b}$ in time $\mathrm{poly}(s, D)^{O(n^\varepsilon)}$.

In Line 3, the map $\mathbf{KI}_{g_\sigma, S}$ can be constructed in time $\exp(O(n^\varepsilon))$ by using Lemma 3.13.

In Line 4, $\tilde{P}(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$ is an $O(n^{2\varepsilon})$-variate polynomial. Using Theorem 4.12, we can factorize $C_{\tilde{P}}(T, \mathbf{KI}_{g, S}(\mathbf{w}), z)$ in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$. For each $j \in [m]$, let $\tilde{G}_j(T, \mathbf{x}, z)$ denote $G_j((T \cdot \mathbf{x}) + (\mathbf{a} \cdot z) + \mathbf{b})$; $\tilde{G}_j$ must be irreducible (this is standard in the literature; for a proof, see [KRSV24, Lemma B.7]).

By Theorem 5.1, for each $j \in [m]$, $\tilde{G}_j(T, \mathbf{KI}_{g, S}(\mathbf{w}), z)$ is an irreducible factor of $\tilde{P}(T, \mathbf{KI}_{g_\sigma, S}(\mathbf{w}), z)$. Thus, for $j \in [m]$, the factorization in Line 4 will compute a circuit $C_{\tilde{G}_j}(T, \mathbf{KI}_{g, S}(\mathbf{w}), z)$ that computes $\tilde{G}_j(T, \mathbf{KI}_{g, S}(\mathbf{w}), z)$, satisfying the property that $\tilde{G}_j(0, \mathbf{KI}_{g, S}(\mathbf{w}), z) = \tilde{G}_j(0, \mathbf{0}, z)$.

By Theorem 9.7, Algorithm 3 correctly computes the list $L' = \{C_{\tilde{G}_1}(T, \mathbf{x}, z), \ldots, C_{\tilde{G}_m}(T, \mathbf{x}, z)\}$ in time $\mathrm{poly}(s, D)^{O(n^\varepsilon)}$. Line 6 computes $C_{G_j}(\mathbf{x}) = C_{\tilde{G}_1}(1, \mathbf{x} - \mathbf{b}, 0)$ for each $G_j(\mathbf{x})$ correctly. Overall, the algorithm takes time $\mathrm{poly}(s, D)^{O(n^\varepsilon)}$ and outputs circuits of size $\mathrm{poly}(s, D)$ for each irreducible factor $G_j(\mathbf{x})$ of $P(\mathbf{x})$. □

9.3   Analysis of Algorithm 1

Theorem 5.2 (Deterministic subexponential time algorithm for factorization of constant-depth circuits). *Fix any $\Delta \in \mathbb{N}$ and $\varepsilon \in (0, 0.5)$. There exists an algorithm $\mathcal{A}_{\Delta, \varepsilon}$ which, for all sufficiently large n,*

- *takes as input a polynomial $P(\mathbf{x}) \in \mathbb{Q}[x_1, \ldots, x_n]$ of degree $D \leq \mathrm{poly}(n)$ with a depth-Δ, size $s \leq \mathrm{poly}(n)$ circuit;*

- *outputs $\mathrm{poly}(s, D)$-sized circuits for each irreducible factor of P, along with the multiplicity of each such factor; and*

- *runs in time $\mathrm{poly}(s, D)^{O(n^{2\varepsilon})}$.*

Proof. Algorithm 1, instantiated with parameters Δ and ε, is the algorithm $\mathcal{A}_{\Delta, \varepsilon}$ claimed in the theorem statement. Theorem 4.4 ensures the correctness of Line 1 so that for each $i \in [r]$, $P_i(\mathbf{x})$ corresponds to the product of irreducible factors of P that have multiplicity i in the factorization (and r is the maximum multiplicity of an irreducible factor). The algorithm from Theorem 4.4 runs in time $\mathrm{poly}(s, D)$ with oracle access to a PIT algorithm for constant-depth circuits; replacing each oracle access by the algorithm in Theorem 3.16 results in a running time of $\mathrm{poly}(s, D)^{O(n^\varepsilon)}$. For each $P_i(\mathbf{x})$, Theorem 9.8 guarantees that the list L_i output by Algorithm 2 in Line 4 is exactly a list of circuits computing the irreducible factors of P_i; moreover, it runs in time $\mathrm{poly}(s, n^D)^{O(n^{2\varepsilon})}$. Since factors of P_i have multiplicity i in P, Line 5 adds the right multiplicity information along with each circuit for factors of P_i. Every irreducible factor of $P(\mathbf{x})$ must occur as an irreducible factor in $P_i(\mathbf{x})$ for some $i \in [r]$; thus, every irreducible factor along with its multiplicity will be included in the output list. □

References

[AF22]   Robert Andrews and Michael A. Forbes. Ideals, determinants, and straightening: proving and using lower bounds for polynomial ideals. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 389–402. ACM, 2022.

[Ale05]   Michael Alekhnovich. Linear Diophantine equations over polynomials and soft decoding of Reed-Solomon codes. *IEEE Trans. Inform. Theory*, 51(7):2257–2265, 2005. (Preliminary version in *43rd FOCS*, 2002).

[AW24]   Robert Andrews and Avi Wigderson. Constant-Depth Arithmetic Circuits for Linear Algebra Problems. In *2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 2367–2386, 2024.

[Bog05]   Andrej Bogdanov. Pseudorandom generators for low degree polynomials. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 21–30. ACM, 2005.

[BSV18]   Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic Factorization of Sparse Polynomials with Bounded Individual Degree. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018*, pages 485–496. IEEE Computer Society, 2018.

[Bü04]   Peter Bürgisser. The Complexity of Factors of Multivariate Polynomials. *Foundations of Computational Mathematics*, 4(4):369–396, September 2004.

[CKS19]   Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure Results for Polynomial Factorization. *Theory of Computing*, 15(13):1–34, 2019.

[DGV24]   Ashish Dwivedi, Zeyu Guo, and Ben Lee Volk. Optimal Pseudorandom Generators for Low-Degree Polynomials Over Moderately Large Fields, 2024. Pre-print available at `arXiv:2402.11915`.

[DL78]   Richard A. DeMillo and Richard J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978.

[DSS22]   Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the Roots: Uniform Closure Results for Algebraic Classes Under Factoring. *J. ACM*, 69(3), June 2022.

[DST24]   Pranjal Dutta, Amit Sinhababu, and Thomas Thierauf. Derandomizing Multivariate Polynomial Factoring for Low Degree Factors, 2024. Pre-print available at `arXiv:2411.17330`.

[For15]   Michael A. Forbes. Deterministic Divisibility Testing via Shifted Partial Derivatives. In *Proceedings of the 2015 IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, FOCS '15, page 451–465, USA, 2015. IEEE Computer Society.

[FS15]   Michael A. Forbes and Amir Shpilka. Complexity Theory Column 88: Challenges in Polynomial Factorization1. *SIGACT News*, 46(4):32–49, dec 2015.

[GK85]   J. von zur Gathen and E. Kaltofen. Factoring Sparse Multivariate Polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985.

[GS99]   Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[Kal89]   Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.

[KI04]   Valentine Kabanets and Russell Impagliazzo. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[KRS23]   Mrinal Kumar, Varun Ramanathan, and Ramprasad Saptharishi. Deterministic Algorithms for Low Degree Factors of Constant Depth Circuits. *CoRR*, abs/2309.09701, 2023. Pre-print available at `arXiv:2309.09701`.

[KRSV24]   Mrinal Kumar, Varun Ramanathan, Ramprasad Saptharishi, and Ben Lee Volk. Towards Deterministic Algorithms for Constant-Depth Factors of Constant-Depth Circuits, 2024. Pre-print available at `arXiv:2403.01965`.

[KS01]   Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 216–223, 2001.

[KSS15]   Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of Polynomial Identity Testing and Polynomial Factorization. *Computational Complexity*, 24(2):295–331, 2015. Preliminary version in the *29th Annual IEEE Conference on Computational Complexity (CCC 2014)*.

[KT90]   Erich Kaltofen and Barry M. Trager. Computing with polynomials given byblack boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990. Computational algebraic complexity editorial.

[Lec07]   Grégoire Lecerf. Improved dense multivariate polynomial factorization algorithms. *Journal of Symbolic Computation*, 42(4):477–494, 2007.

[LLL82]   Arjen K. Lenstra, Hendrik W. Lenstra Jr., and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

[LST21]   Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2021)*, pages 804–814. IEEE, 2021. Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR21-081.

[NW94]   Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. Available on `citeseer:10.1.1.83.8416`.

[Ore22]   Øystein Ore. Über höhere Kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

[Sap15]   Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015.

[Sch80]   Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980.

[ST20]   Amit Sinhababu and Thomas Thierauf. Factorization of Polynomials Given By Arithmetic Branching Programs. In *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:19, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[Str73]   Volker Strassen. Vermeidung von Divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

[Sud97]   Madhu Sudan. Decoding of Reed Solomon Codes beyond the Error-Correction Bound. *J. Complexity*, 13(1):180–193, 1997.

[Sud98]   Madhu Sudan. Lecture notes for the course 'Algebra and Computation', 1998. Available from `http://people.csail.mit.edu/madhu/FT98/`.

[SV10]   Amir Shpilka and Ilya Volkovich. On the Relation between Polynomial Identity Testing and Finding Variable Disjoint Factors. In *Automata, Languages and Programming, 37th International Colloquium, ICALP 2010, Bordeaux, France, July 6-10, 2010, Proceedings, Part I*, volume 6198 of *Lecture Notes in Computer Science*, pages 408–419. Springer, 2010.

[SW01]   Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001. Preliminary version in the *14th Annual IEEE Conference on Computational Complexity (CCC 1999)*.

[SY10]   Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[Vol17]   Ilya Volkovich. On Some Computations on Sparse Polynomials. volume 81 of *LIPIcs*, pages 48:1–48:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[vzGG13]   Joachim von zur Gathen and Jürgen Gerhard. *Modern Computer Algebra*. Cambridge University Press, 3 edition, 2013.

[Zip79]   Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.