Factors of Low Individual Degree Polynomials*

Rafael Oliveira

Department of Computer Science, Princeton University
35 Olden St., Princeton NJ, USA
rmo@cs.princeton.edu

Abstract

In [8], Kaltofen proved the remarkable fact that multivariate polynomial factorization can be done efficiently, in randomized polynomial time. Still, more than twenty years after Kaltofen's work, many questions remain unanswered regarding the complexity aspects of polynomial factorization, such as the question of whether factors of polynomials efficiently computed by arithmetic formulas also have small arithmetic formulas, asked in [10], and the question of bounding the depth of the circuits computing the factors of a polynomial.

We are able to answer these questions in the affirmative for the interesting class of polynomials of bounded individual degrees, which contains polynomials such as the determinant and the permanent. We show that if $P(x_1, \ldots, x_n)$ is a polynomial with individual degrees bounded by r that can be computed by a formula of size s and depth d, then any factor $f(x_1, \ldots, x_n)$ of $P(x_1, \ldots, x_n)$ can be computed by a formula of size $\mathsf{poly}((rn)^r, s)$ and depth $d+5$. This partially answers the question above posed in [10], that asked if this result holds without the exponential dependence on r. Our work generalizes the main factorization theorem from Dvir et al. [2], who proved it for the special case when the factors are of the form $f(x_1, \ldots, x_n) \equiv x_n - g(x_1, \ldots, x_{n-1})$. Along the way, we introduce several new technical ideas that could be of independent interest when studying arithmetic circuits (or formulas).

1998 ACM Subject Classification F.1.1 Models of Computation, F.1.3 Complexity Measures and Classes, F.2.1 Numerical Algorithms and Problems

Keywords and phrases Arithmetic Circuits, Factoring, Algebraic Complexity

Digital Object Identifier 10.4230/LIPIcs.CCC.2015.198

1 Introduction

Let $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a multivariate polynomial over a field $\mathbb{F}$. The individual degree of f with respect to variable x_i, denoted by $\deg_{x_i}(f)$, is the largest power of x_i appearing in a monomial of f. Many interesting polynomials have bounded individual degree, such as the Permanent and Determinant polynomials. Moreover, the class of polynomials of bounded individual degree is closed under factorization, since if a polynomial $f(x_1, \ldots, x_n)$ has individual degrees bounded by r, so will its factors. In this work, we study the problem of formula (circuit) factorization of polynomials of low individual degree.

One of the basic operations on polynomials is factorization. This problem can be phrased as follows: given a polynomial $P(x_1, \ldots, x_n)$, decide whether $P(x_1, \ldots, x_n)$ is irreducible, or if not, output one of its factors, which we denote by $f(x_1, \ldots, x_n)$. From the computational perspective, we will usually be given a device computing the polynomial P, and we will be asked to output a similar device computing f. In the field of arithmetic complexity, the most natural device for computing polynomials is an arithmetic circuit or a formula (see

* Research supported by NSF grant CCF-1217416 and by the Sloan fellowship.

30th Conference on Computational Complexity (CCC'15).
Editor: David Zuckerman; pp. 198–216

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Definition 1.1 below). Therefore, we will assume that we are given P as an arithmetic circuit (formula) and output one of its factors in the same representation. We now give the definition of an arithmetic circuit/formula:

▶ **Definition 1.1.** An *arithmetic circuit* Γ is a directed acyclic labeled graph in which the vertices are called 'gates'. The gates of Γ with in-degree 0 are called *inputs* and are labeled by either a variable from $\{x_1, \ldots, x_n\}$ or by a field element from $\mathbb{F}$. Every other gate of Γ is labeled by either '$\times$' or '$+$' and has in-degree 2. (If we talk about bounded depth circuits/formulas, then we remove the restriction on the in-degree.) There is one gate with out-degree 0, which we call the *output gate*. Each gate in Γ computes a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ in the natural way. An arithmetic circuit is called a *formula* if its underlying graph is a tree. The *size* of a circuit (formula) Γ, written $|\Gamma|$, is given by the number of edges in the circuit (formula) and the *depth* of Γ, written $\text{depth}(\Gamma)$, is defined as the length of the longest directed path in the graph of Γ.

Polynomial factorization is one of the cornerstone problems in modern computer algebra, and as such has been the focus of intensive research. The past three decades have seen major advances on the development of efficient algorithms for polynomial factorization, pioneered by the works of Lenstra et al. and Kaltofen [11, 7, 8, 9]. In addition to the general problem, polynomial factorization has also been studied in many other important (and more restricted) representations. For instance, in the sparse representation, where the input polynomial is given as a list of its coefficients and monomials, the works of Lenstra, Kaltofen and von zur Gathen [12, 4] give efficient algorithms for sparse factorization in the univariate and in the multivariate cases. For a more complete survey on polynomial factorization we refer the reader to the survey [9] and to the book [3].

In the seminal work of Kaltofen [8], it is proved that if $P(x_1, \ldots, x_n)$ of total degree D can be computed by an arithmetic circuit of size s, then any of its factors have arithmetic circuits of size $\mathsf{poly}(n, s, D)$. Moreover, Kaltofen gives a randomized algorithm that with high probability outputs such a factor in polynomial time. This result, besides settling an important complexity theoretic question, has since then had a great impact in many areas of computer science, such as coding theory [16, 5], derandomization [6] and cryptography [1]. However, many interesting questions on the complexity of arithmetic circuits or formulas under factorization remain unanswered. In particular, we study the following two questions, where the first one was asked in the work of Kopparty et al. [10], while the second question was stated as an open problem in the survey [15, Open Problem 19]:

1. If $P(x_1, \ldots, x_n)$ of total degree D is computed by an arithmetic formula of size s, is it true that any of its factors will also have formulas of size $\mathsf{poly}(n, s, D)$?
2. If $P(x_1, \ldots, x_n)$ can be computed by a circuit of size s and depth d, can its factors be computed by a circuit of size $\mathsf{poly}(s)$ and depth $O(d)$?

In this work, we answer both of these questions in the affirmative, in the case where the input polynomial P has bounded individual degrees. In particular, we show:

▶ **Theorem 1.2.** *Let $P(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n] \setminus \{0\}$ be such that $\deg_{x_i}(P) \leq r$, $1 \leq i \leq n$, and let $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a factor of P, where $\mathbb{F}$ is a field of characteristic zero. If there exists a formula (circuit) of size s and depth d computing P, then there exists a formula (circuit) of depth $d+5$ and size $\mathsf{poly}((nr)^r, s)$ that computes $f(x_1, \ldots, x_n)$. Moreover, if we require the in-degree of each gate to be 2, then the size remains the same and the depth becomes $d + O(r \log(nr))$.*

Notice that our theorem has no restriction on the individual degrees of the polynomials computed by the intermediate gates of the circuit (that is, we have no syntactic restrictions). We only care about the individual degrees of the output polynomial, which we regard as bounded by a constant, denoted by r, in the theorem above.

Theorem 1.2 provides a direct answer to the second question posed above in the case where P has bounded individual degrees (that is, r is a constant). The connection between Theorem 1.2 and the first question comes from the fact that one can always balance formulas to have logarithmic depth. More precisely, suppose that we are given a formula Φ (with in-degree bounded by 2) of size $s = \mathsf{poly}(n)$ computing P. By Theorem 2.7 in [15], we can assume that Φ is of size $\mathsf{poly}(s)$ and $\mathrm{depth}(\Phi) = O(\log s)$. Hence, Theorem 1.2 implies that there exists a formula Ψ, with in-degree bounded by 2, of depth $\mathrm{depth}(\Psi) = \mathrm{depth}(\Phi) + O(r \log(sn)) = O(\log s)$ and size $\mathsf{poly}((nr)^r, s) = \mathsf{poly}(s)$ computing any factor $f(x_1, \ldots, x_n)$ of P. This provides an affirmative answer to the first question.

Before giving an overview of the proof of Theorem 1.2, we give some background on related work on factorization in general and in bounded depth circuits.

The problem of factoring in bounded depth was studied previously in [2], who showed that if $P(x_1, \ldots, x_n)$ has a depth d circuit of size s and $\deg_{x_n}(P) \leq r$, then its factors of the form $x_n - \phi(x_1, \ldots, x_{n-1})$ have depth $d + 3$ circuits of size $\mathsf{poly}(n^r, s)$. This result was used to extend the hardness-randomness tradeoffs of [6] to the bounded depth model. Our main theorem generalizes their result to any factor of P, provided that P has bounded individual degrees.

Shpilka and Volkovich in [14] initiated the study of factorization of multilinear polynomials, which are the most basic case of polynomials of bounded individual degrees. They relate the problem of deterministically factoring multilinear polynomials to the problem of performing deterministic Polynomial Identity Testing (PIT). In their paper, they prove that these two problems are roughly equivalent in the multilinear setting for most restricted multilinear circuit classes that have been studied. Since the problem of performing deterministic PIT seems to be hard, even for the class of multilinear formulas, this shed some light on the difficulty of obtaining deterministic factorization even for this model. This equivalence between deterministic PIT and deterministic polynomial factorization was later generalized by Kopparty et al. in [10] to polynomials (of polynomial degree) computed by general circuits. Since we prove here that, for polynomials of bounded individual degrees computed by circuits of small depth, their factors can also be computed by circuits of small depth, one could hope for similar connections between PIT for restricted classes of circuits – say of bounded depth and low individual degrees – and factorization of polynomials in such classes.

2 Proof Overview

In this section, we give an overview of the proof of the main theorem. For simplicity of exposition, we will only refer to arithmetic circuits in this overview, but our results hold true for formulas as well, as the statements in the later sections show. We begin with a definition:

▶ **Definition 2.1** (Approximate Root)**.** Let $P(x_1, \ldots, x_n, y)$ be a polynomial in $\mathbb{F}[x_1, \ldots, x_n, y]$. We say that $q(x_1, \ldots, x_n)$ is a *root* of P *up to degree* t if all the homogeneous parts up to degree t of the polynomial $P(x_1, \ldots, x_n, q(x_1, \ldots, x_n))$ are zero. That is, $P(x_1, \ldots, x_n, q(x_1, \ldots, x_n))$ only has monomials of degree larger than t.

Given a polynomial $P(x_1, \ldots, x_n, y) \in \mathbb{F}[x_1, \ldots, x_n, y]$ with individual degree in y bounded by r, Dvir et al. [2] show that if $P(0, \ldots, 0, y)$ has no double roots, that is, $P(0, \ldots, 0, y)$ can

be factored as

$$P(0,\dots,0,y) \equiv c \cdot \prod_{i=1}^{r}(y-\mu_i)$$

where $\mu_i \neq \mu_j$ for $i \neq j$, then for each μ_i, there exists an approximate root $q_{i,t}(x_1,\dots,x_n)$ of P up to degree t such that $q_{i,t}(0,\dots,0) = \mu_i$. Moreover, they show that if P is computed by a circuit Γ of size s and depth d, then there exists a circuit of size $\mathsf{poly}(t^r, s)$ and depth $d+2$ computing $q_{i,t}(x_1,\dots,x_n)$.

With this idea in mind, suppose for simplicity that

$$P(x_1,\dots,x_n,y) \equiv \prod_{i=1}^{r}(y-g_i(x_1,\dots,x_n)),$$

where each polynomial $g_i(x_1,\dots,x_n)$ has a nonzero constant term μ_i and $\mu_i \neq \mu_j$ for $i \neq j$. In this case we are in the framework of [2], since

$$P(0,\dots,0,y) \equiv \prod_{i=1}^{r}(y-\mu_i)$$

and the roots μ_i are distinct. As Section 4 shows, we can guarantee distinct roots in $P(0,\dots,0,y)$ by using a random shift of the variables $(x_1,\dots,x_n)$, as long as P is square-free. Therefore, for each μ_i and $t \geq 1$, we can find polynomials $q_{i,t}(x_1,\dots,x_n)$ such that $q_{i,t}(0,\dots,0) = \mu_i$ and the polynomial $P(x_1,\dots,x_n,q_{i,t}(x_1,\dots,x_n))$ only has terms of degree larger than t. Since

$$P(x_1,\dots,x_n,q_{i,t}(x_1,\dots,x_n)) \equiv \prod_{j=1}^{r}(q_{i,t}(x_1,\dots,x_n)-g_j(x_1,\dots,x_n)),$$

the minimum degree terms of $P(x_1,\dots,x_n,q_{i,t}(x_1,\dots,x_n))$ must come from the product of the minimum degree terms of each of the polynomials $q_{i,t}(x_1,\dots,x_n)-g_j(x_1,\dots,x_n)$. Notice that for each $j \neq i$, the constant term of $q_{i,t}(x_1,\dots,x_n)-g_j(x_1,\dots,x_n)$ is equal to $\mu_i-\mu_j$, which is nonzero. Therefore, the minimum degree terms of $P(x_1,\dots,x_n,q_{i,t}(x_1,\dots,x_n))$ must come from the minimum degree terms of the polynomial $q_{i,t}(x_1,\dots,x_n)-g_i(x_1,\dots,x_n)$. Because $P(x_1,\dots,x_n,q_{i,t}(x_1,\dots,x_n))$ only has terms of degree larger than t, the same must happen to the polynomial $q_{i,t}(x_1,\dots,x_n)-g_i(x_1,\dots,x_n)$. This implies that $q_{i,t}(x_1,\dots,x_n)$ approximates the actual root $g_i(x_1,\dots,x_n)$ of P up to degree t. Hence, if we pick t larger than the total degree of g_i, the lower degree terms of $q_{i,t}$ correspond to the root g_i, and therefore we can recover this root g_i (and use it to factor P).

There are two main issues with this approach that we need to overcome, if we are to generalize it. The first issue is that P may not factor into linear factors in y, that is, polynomials of the form $y-g_i(x_1,\dots,x_n)$. The second one is that P need not be monic in y, in which case we will still need to recover its leading coefficient – which is a polynomial in $\mathbb{F}[x_1,\dots,x_n]$.

To deal with the first issue, let us study a toy example: assume that P is monic in y with $\deg_y(P) = r$, that is,

$$P(x_1,\dots,x_n,y) \equiv y^r + \sum_{i=0}^{r-1} P_i(x_1,\dots,x_n)y^i,$$

but P does not factor into linear factors in y. Let $f(x_1,\dots,x_n,y)$ be one of its factors, of degree k in y. Since P is monic in y, we know that f must also be monic in y. Note that

if we work over the algebraic closure of $\mathbb{F}(x_1, \ldots, x_n)$ (that is, the field $\overline{\mathbb{F}(x_1, \ldots, x_n)}$), we can factor P (and f) into linear factors in y. In this work, we will not describe what the algebraic closure of $\mathbb{F}[x_1, \ldots, x_n]$ is, since it is a very complex field, and it is not needed in our proof. We only mention $\overline{\mathbb{F}(x_1, \ldots, x_n)}$ here to give us some intuition on how to generalize the root finding approach described above. For simplicity, think of elements of the closure as "functions" over the variables $x_1, \ldots, x_n$. Since f divides P, if

$$P(x_1, \ldots, x_n, y) \equiv \prod_{i=1}^{r} (y - \varphi_i(x_1, \ldots, x_n)),$$

then there will be indices (say i from 1 to k) such that

$$f(x_1, \ldots, x_n, y) \equiv \prod_{i=1}^{k} (y - \varphi_i(x_1, \ldots, x_n)).$$

However, it is worth noting that these linear factors will not be polynomials! Nevertheless, the fact that they share some roots in the closure of $\mathbb{F}[x_1, \ldots, x_n]$ gives us a hint on what to do next. To overcome this problem, we will (in Lemma 6.1 and Corollary 6.2) approximate these functions φ_i by polynomials $g_{i,t}$, in a way that the polynomial

$$g_t(x_1, \ldots, x_n, y) \equiv \prod_{i=1}^{k} (y - g_{i,t}(x_1, \ldots, x_n))$$

agrees with f on the terms of order smaller than t. Therefore, for large enough t, the lower order terms of $g_t(x_1, \ldots, x_n, y)$ will correspond to the polynomial f, which we can then obtain by interpolation (Lemma 3.3). We can think of each polynomial $g_{i,t}$ as the Taylor expansion of φ_i up to degree t.

The way we obtain these approximations to the roots (the polynomials $g_{i,t}$) is by a procedure similar in nature to Hensel lifting. Suppose that $\varphi_i(0, \ldots, 0) = \mu_i$ for $1 \leq i \leq k$, and moreover, suppose that $\mu_i \neq \mu_j$ for $i \neq j$. From each valuation μ_i, we will construct a family of polynomials $g_{i,t}$ of degree t, such that $g_{i,t}(x_1, \ldots, x_n)$ is a root of f up to degree t. Now, the question is: how can we construct this family of polynomials if we do not have access to f? The answer to this question lies on the fact that each root $y - \varphi_i$ of f is also a root of P, and therefore we can access the valuations of φ_i's through the circuit computing P. Hence, we will use the fact that the φ_i's are also roots of P in order to find the polynomials g_t that approximate f (Lemma 7.1).

To overcome the second main issue, that the polynomial P may not be monic, let us define

$$f(x_1, \ldots, x_n, y) \equiv \sum_{i=0}^{k} f_i(x_1, \ldots, x_n) y^i \quad \text{and} \quad P(x_1, \ldots, x_n, y) \equiv \sum_{i=0}^{r} P_i(x_1, \ldots, x_n) y^i,$$

where $f_k(x_1, \ldots, x_n) \not\equiv 0$ and $P_r(x_1, \ldots, x_n) \not\equiv 0$. If f divides P, then it must be the case that the leading coefficient f_k of f divides the leading coefficient P_r of P. Hence, a possible solution to this second issue would be to find, by some kind of induction, a small circuit for f_k based on the circuit for P_r that we obtain from P. Then, we could generalize the factoring result for monic polynomials to the case where the factors are rational functions of the form

$$\frac{f(x_1, \ldots, x_n, y)}{f_k(x_1, \ldots, x_n)} \equiv y^k + \sum_{i=0}^{k-1} \frac{f_i(x_1, \ldots, x_n)}{f_k(x_1, \ldots, x_n)} y^i.$$

With these two results, we could multiply the circuits computing f_k and $\frac{f}{f_k}$ to obtain our factor f.

More precisely, if we could find, by induction on the number of variables, a small circuit Φ_k for f_k based on the circuit Γ_r for P_r that we obtain from P via interpolation (Lemma 3.4), and if we could find a small circuit Υ for the rational function $\frac{f}{f_k}$ based on the circuit Γ computing P (Lemma 7.1), then the circuit given by $\Upsilon \times \Phi_k$ would compute the polynomial f, as we wanted.

One problem with this approach is that, even if we can generalize the monic factoring result to monic rational functions as above, as far as we know, the best bound on the size of the circuit Γ_r computing P_r is given by $3r \cdot s$ (see Lemma 3.4). Therefore, if we define $T(n, s)$ as the maximum size of a factor of a polynomial in n variables computed by a circuit of size s, the induction given by the procedure above would give us the following bounds on the size:

$$T(n+1, s) \leq T(n, 3r \cdot s) + \mathsf{poly}((nr)^r, s).$$

The reason for this bound is the following: $P(x_1, \ldots, x_n, y)$ has $n+1$ variables and is computed by Γ, which has size s. Hence, the maximum size of a factor f is by definition $T(n+1, s)$. Since f_k divides the leading coefficient P_r, which is computed by Γ_r of size $3rs$ and has n variables, the bound we have on the size of Φ_k is given by $T(n, 3rs)$, because now the input polynomial is P_r. Assuming that the size of f/f_k can be bounded by $((nr)^r \cdot s)^\alpha$, for some constant α (which we can by Lemma 7.1), we obtain the additive factor $\mathsf{poly}((nr)^r, s)$. Since the circuit for f is given by $\Upsilon \times \Phi_k$, we need to add the bounds on the sizes for Φ_k and Υ. However, when we solve this equation, we obtain that

$$T(n+1, s) \leq T(1, (3r)^n \cdot s) + \mathsf{poly}((nr)^r, (3r)^n \cdot s)$$

which is exponential in n, the number of variables! Therefore, this approach, as it is, cannot work.

The main problem with the recursion above is that the bound on the circuit size of the leading coefficient, if we only use Lemma 3.4, keeps getting worse as we reduce the number of variables – it will become $(3r)^\ell \cdot s$ if we get rid of ℓ variables. To get around this issue, we define the *reversal* of a polynomial with respect to a specific variable and we study its properties with regards to divisibility. If

$$P(x_1, \ldots, x_n, y) \equiv \sum_{i=0}^{r} P_i(x_1, \ldots, x_n) y^i$$

is a polynomial, with $P_r(x_1, \ldots, x_n) \cdot P_0(x_1, \ldots, x_n) \not\equiv 0$, we define its reversal with respect to y as the polynomial

$$\tilde{P}(x_1, \ldots, x_n, y) \equiv \sum_{i=0}^{r} P_i(x_1, \ldots, x_n) y^{r-i}.$$

That is, $\tilde{P}$ is obtained from the polynomial P by "reversing" the coefficients $P_i(x_1, \ldots, x_n)$. It is easy to see that f divides P iff $\tilde{f}$ divides $\tilde{P}$. By performing a reversal, notice that we have transformed the leading coefficient of our problem from $P_r(x_1, \ldots, x_n)$ to $P_0(x_1, \ldots, x_n)$. This has the advantage that now, the leading coefficient of our input polynomial can be computed by the circuit $\Gamma|_{y=0}$ (that is, the circuit obtained from Γ by setting $y = 0$), which has size $\leq s$. This now allows us to recurse into the division of f_0 by P_0 (the new leading

coefficients after the reversal) without paying the multiplicative cost on the size of the circuit. Hence with this idea we avoid paying the exponential blowup on the circuit size! On the coin side, notice that the size of the circuit computing the polynomial $\tilde{P}$ is bounded by $8r^2 \cdot s$, according to Lemma 3.7. But this blow up does not hurt us, since the reversal is not cumulative.

More precisely, we now have the following recursion: we want to bound the size of a factor of P, computed by a circuit Γ of size s and on $n+1$ variables. This bound is by definition $T(n+1, s)$. Let $\widetilde{\Gamma}$ be a circuit computing $\tilde{P}$. Suppose we can find a circuit computing f/f_0 of size bounded by $((nr)^r \cdot |\widetilde{\Gamma}|)^\alpha \leq ((nr)^r \cdot 8r^2 s)^\alpha$, for some constant α (which we can by Lemma 7.1). Then we are only left with the problem of finding a small circuit for f_0, which divides P_0, which in turn can be computed by a circuit of size bounded by s in n variables. The bound for a circuit for f_0 is given in this case by $T(n, s)$, by definition of the function T. Therefore, our recursion becomes

$$T(n+1, s) \leq T(n, s) + ((nr)^r \cdot 8r^2 \cdot s)^\alpha$$

which implies that

$$T(n, s) \leq n \cdot ((nr)^r \cdot 8r^2 \cdot s)^\alpha = \mathsf{poly}((nr)^r, s),$$

as we wanted!

The idea of the reversal of a polynomial is similar to the definition of *reversal* of a univariate polynomial given in [3, §9.1]. This notion of reversal is used there to perform division with remainder for univariate polynomials by using Newton iteration.

To generalize the monic factoring result to the case when f is monic in y with rational coefficients, we introduce the idea of an approximation polynomial of a rational function (see Section 5), and we use this approximation polynomial in Lemma 7.1 (instead of the rational function) as the "factor" of the input polynomial. If f is a rational function of the form

$$f(x_1, \ldots, x_n, y) \equiv \frac{1}{1 - g(x_1, \ldots, x_n)} \cdot \sum_{i=0}^{k} f_i(x_1, \ldots, x_n) y^i,$$

where $g(x_1, \ldots, x_n)$ and $f_i(x_1, \ldots, x_n)$ are polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ such that $g(0, \ldots, 0) = 0$, we define its approximation polynomial (to degree m) as the following polynomial

$$\psi_{f,m}(x_1, \ldots, x_n, y) \equiv (1 + g + g^2 + \ldots + g^m) \cdot \sum_{i=0}^{k} f_i y^i,$$

where $g \equiv g(x_1, \ldots, x_n)$ and $f_i \equiv f_i(x_1, \ldots, x_n)$. This polynomial "approximates" the rational function $f(x_1, \ldots, x_n, y)$ in the sense that, for large enough m, the polynomial obtained by $\psi_{f,m}(x_1, \ldots, x_n, y) \cdot (1 - g(x_1, \ldots, x_n))$ is equal to $f(x_1, \ldots, x_n) \cdot (1 - g(x_1, \ldots, x_n))$, up to high order terms (see Observation 5.3), which we can get rid of by interpolation (Lemma 3.3). By adapting the approach in [2] to work with approximation polynomials, we can find all the "roots" of the approximation polynomials, and after that combine this approximation polynomial with the circuit obtained to compute the leading term.

After we take care of finding the leading coefficient $f_0(x_1, \ldots, x_n)$ (of the reversed polynomial $\tilde{f}(x_1, \ldots, x_n, y)$), and after recovering the approximation polynomial $\psi_{\tilde{f},m}$ (see Lemma 7.1), we can multiply it by f_0 to obtain the factor f (up to high order terms) which, after interpolation, becomes our desired factor (see Theorem 7.2).

We conclude this proof outline with a basic roadmap of the main ideas involved in this work:

1. Given a circuit Γ for our polynomial $P(x_1, \ldots, x_n, y)$, we find a circuit $\tilde{\Gamma}$ computing the reversal polynomial $\tilde{P}(x_1, \ldots, x_n, y)$. (Lemma 3.7)
2. We use the circuit $\tilde{\Gamma}$ to find small circuits $\Phi_{i,t}$ for each approximate root of $\tilde{P}$ up to degree t. (Section 6)
3. Since $\tilde{f}$, divides $\tilde{P}$ (Lemma 3.8), any approximate root of $\tilde{f}$ will also be an approximate root of $\tilde{P}$. By combining the circuits $\Phi_{i,t}$ computing the approximate roots of $\tilde{f}(x_1, \ldots, x_n, y)$, find circuit Ψ computing the approximation polynomial (see Section 5) of the monic rational function $\dfrac{\tilde{f}(x_1, \ldots, x_n, y)}{f_0(x_1, \ldots, x_n)}$. (Lemma 7.1)
4. By induction, obtain the circuit Λ_0 computing $f_0(x_1, \ldots, x_n)$, through the circuit $\Gamma|_{y=0}$ computing $P_0(x_1, \ldots, x_n) \equiv P(x_1, \ldots, x_n, 0)$.
5. We then prove that the lower order terms of the circuit $\Phi = \Lambda_0 \times \Psi$ compute the polynomial $\tilde{f}$. (Theorem 7.2)
6. By interpolation (Lemma 3.3) and by the Reversal Lemma (Lemma 3.7), obtain the lower order terms from Φ computing f.

2.1 Organization

The rest of the paper is organized as follows: in Section 3, we set up notations, go over some useful background and discuss the concept of reversal of a polynomial. In Section 4, we introduce the concept of properly splitting variable restrictions. In Section 5, we formally introduce the concepts of standard forms and approximation polynomials. In Section 6, we adapt the approach of [2] to find small formulas for the roots of $P(x_1, \ldots, x_n, y)$. In Section 7 we prove our main technical lemma and theorem. In Section 8, we conclude and propose some open problems.

For the sake of brevity of exposition, we only give a proof of our main technical theorem. The proofs of all other facts stated in this paper can be found in the full version [13].

3 Preliminaries

In this section, we establish the notation that will be used throughout the paper and some technical background that will be needed in the proof of our main theorem.

3.1 Notations

From this point on, we will use boldface for vectors, and regular font for scalars. Thus, we will denote the vector $(x_1, \ldots, x_n)$ by $\mathbf{x}$. If we want to multiply the vector $\mathbf{x}$ by a scalar z we will denote this product by $z\mathbf{x}$.

We will denote our base field by $\mathbb{F}$, assume that $\mathbb{F}$ has characteristic zero and that it is algebraically closed. The results in this paper also hold for non-closed fields of large enough characteristic, if we allow ourselves to use elements from field extensions. The assumptions just made are for clarity of exposition.

Let $\mathbb{N}_0$ be the set of natural numbers including zero, that is, $\mathbb{N}_0 = \{0, 1, 2, 3, \ldots\}$. If $\mathbf{e} \in \mathbb{N}_0^n$ is a vector of natural numbers and $\mathbf{x} = (x_1, \ldots, x_n)$ is a vector of formal variables, we define $\mathbf{x}^{\mathbf{e}} = \prod_{i=1}^{n} x_i^{e_i}$. That is, $\mathbf{x}^{\mathbf{e}}$ is the monomial corresponding to the product of the variables $\prod_{i=1}^{n} x_i^{e_i}$, where each variable is raised to the proper power.

We will denote $\mathbb{F}(\mathbf{x})[y]$ as the set of polynomials in the variable y whose coefficients are rational functions on the variables $\mathbf{x}$. That is, $f(\mathbf{x}, y) \in \mathbb{F}(\mathbf{x})[y]$ iff it can be expressed in the form $f(\mathbf{x}, y) \equiv \sum_{i=0}^{k} \frac{f_i(\mathbf{x})}{g_i(\mathbf{x})} y^i$, with $f_i(\mathbf{x}), g_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}], 0 \leq i \leq k$.

When working with a polynomial in $\mathbb{F}[\mathbf{x}, y]$, we might be interested in looking at the homogeneous parts of a polynomial with respect to certain variables only. This will be particularly useful when lifting the "roots" of a polynomial $f(\mathbf{x}, y)$ of the form $y - q(\mathbf{x})$ in order to obtain a circuit computing $f(\mathbf{x}, y)$. To this end, we introduce the following definition.

▶ **Definition 3.1** (Partial Homogeneous Parts)**.** Let $P(\mathbf{x}, y) \equiv \sum_{\mathbf{d}} \alpha_{\mathbf{d}}(y) \cdot \mathbf{x}^{\mathbf{d}}$ be a polynomial in $\mathbb{F}[\mathbf{x}, y]$, where each $\alpha_{\mathbf{d}}(y) \in \mathbb{F}[y]$. For each $m \in \mathbb{N}_0$, we define $H^{\mathbf{x}}_m[P]$ as the polynomial formed by the homogeneous parts of degree m of $P(\mathbf{x}, y)$, when seen as a polynomial in $\mathbb{F}[y][\mathbf{x}]$, that is, when considered as a polynomial on the variables $\mathbf{x}$, and regarding y as a constant. More explicitly, $H^{\mathbf{x}}_m[P]$ is equal to the sum of all monomials of P that have degree m in $x_1, \ldots, x_n$, without any restrictions on the degree of y. We also define $H^{\mathbf{x}}_{\leq m}[P] \equiv \sum_{i=0}^{m} H^{\mathbf{x}}_i[P]$.

For example, if $P(\mathbf{x}, y) \equiv (x_1x_3x_4 - x_2^3 + x_1x_2)y^2 + (x_1^2x_3 - x_4)y + x_2^2x_3 - x_1x_4$, we have that $H^{\mathbf{x}}_3[P(\mathbf{x}, y)] \equiv (x_1x_3x_4 - x_2^3)y^2 + x_1^2x_3y + x_2^2x_3$.

Notice that if $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} P_i(\mathbf{x})y^i$, then the partial homogeneous parts satisfy the following property:

$$H^{\mathbf{x}}_m[P(\mathbf{x}, y)] \equiv \sum_{i=0}^{r} H^{\mathbf{x}}_m[P_i(\mathbf{x})] \cdot y^i.$$

Therefore, this definition of partial homogeneous parts agrees with the definition of homogeneous parts if $P(\mathbf{x}, y)$ does not depend on variable y.

When talking about partial homogeneous parts of a polynomial, it is useful to have a notion of minimum degree with respect to some variables.

▶ **Definition 3.2** (Minimum Degree)**.** Let $f(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial. We define $\text{mindeg}_{\mathbf{x}}(f(\mathbf{x}, y))$ to be the minimum degree of polynomial $f(\mathbf{x}, y)$ on the variables $\mathbf{x}$. In other words, we have $\text{mindeg}_{\mathbf{x}}(f(\mathbf{x}, y)) = \min_\ell \left(H^{\mathbf{x}}_\ell[f] \not\equiv 0\right)$. For instance, if $f(\mathbf{x}, y) = x_1x_2x_3y - x_1^2x_2^2 + x_3^5$, we have that $\text{mindeg}_{\mathbf{x}}(f) = 3$.

3.2 Basic Operations on Circuits and Formulas

We begin with the following standard lemma on obtaining the homogeneous components of a polynomial. The version below is from [2].

▶ **Lemma 3.3** (Homogeneous Components Through interpolation)**.** *Let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a polynomial with degree $\deg(P) = m$ such that P can be computed by a formula (circuit) Γ of depth d. Then, there exists a formula (circuit) Δ with $m + 1$ outputs, of size $|\Delta| \leq 9m^2 \cdot |\Gamma|$ and depth $\text{depth}(\Delta) \leq \text{depth}(\Gamma) + 1$ that computes $H^{\mathbf{x}}_0[P], \ldots, H^{\mathbf{x}}_m[P]$. Moreover, if the topmost gate in the formula (circuit) for $P(\mathbf{x})$ is an addition gate, then we have $\text{depth}(\Delta) = \text{depth}(\Gamma) = d$.*

The next lemma shows us how to obtain the coefficients of a polynomial through interpolation.

▶ **Lemma 3.4** (Interpolation). *Let* $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^i P_i(\mathbf{x})$ *be a polynomial computed by a formula (circuit)* Γ*. Then for each* $i \in \{0, 1, \ldots, r\}$*, there exists a formula (circuit)* Φ_i *such that* $|\Phi_i| \leq 3r \cdot |\Gamma|$ *and* Φ_i *computes the polynomial* $P_i(\mathbf{x})$*.*

Given an irreducible polynomial $g(\mathbf{x}, y)$ and a polynomial $P(\mathbf{x}, y)$ that is divisible by g, it will be useful for us to find a polynomial $D(\mathbf{x}, y)$ that is divisible by g and it is also square-free with respect to g, that is, $g(\mathbf{x}, y) \nmid \frac{\partial D}{\partial y}(\mathbf{x}, y)$. The next lemma shows that we can find such a polynomial efficiently.

▶ **Lemma 3.5.** *Let* $g(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *be an irreducible polynomial that divides a polynomial* $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$*, where* $\deg_y(P) \leq r$ *and let* Γ *be a formula computing* $P(\mathbf{x}, y)$*. Then, there exists a formula* Δ *that computes a polynomial* $D(\mathbf{x}, y)$ *such that* $g(\mathbf{x}, y) \mid D(\mathbf{x}, y)$*,* $g(\mathbf{x}, y) \nmid \dfrac{\partial D}{\partial y}(\mathbf{x}, y)$*,* $|\Delta| \leq 9r^2 \cdot |\Gamma|$ *and* $\mathrm{depth}(\Delta) \leq \mathrm{depth}(\Gamma)$*. Moreover, the output gate of* Δ *is an addition gate and for each variable* $z \in \{\mathbf{x}, y\}$*, we have that* $\deg_z(D) \leq \deg_z(P)$*.*

The following observation will be very useful to convert small depth formulas into formulas with fanin bounded by 2.

▶ **Observation 3.6.** *Any formula* Φ *of size* s *and depth* d*, without restrictions on the fanin of any of its gates, can be computed by a formula* Ψ *of size* $2s$ *and depth* $d \cdot (1 + \log(s))$*, where each gate has fanin* 2*.*

To see that this observation is true, just replace each addition (multiplication) gate of fanin t by a balanced formula of size $2t$ made only with addition (multiplication) gates. Since $t \leq s$, and a balanced formula of size $2t$ has depth $1 + \log t$, we have that each gate will be replaced by a formula of depth at most $1 + \log s$. The replacement by a balanced formula clearly does not change the computation, and the depth increases by a multiplicative factor of $1 + \log s$, as we wanted.

3.3 Reversal of Polynomials

In this section, we define a very useful operation for polynomials, which serves as a crucial tool in the proof of our main theorem. This operation, which we call *reversal*, simply maps a polynomial $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} P_i(\mathbf{x}) y^i$, with $P_r(\mathbf{x}) \cdot P_0(\mathbf{x}) \not\equiv 0$, to $\tilde{P}(\mathbf{x}) \equiv \sum_{i=0}^{r} P_i(\mathbf{x}) y^{r-i}$.

The restriction that $P_r(\mathbf{x}) \cdot P_0(\mathbf{x}) \not\equiv 0$ is needed in this paper because it preserves irreducibility, as we will see in Lemma 3.8 and Corollary 3.9. We begin by showing that the reversal can be computed almost as efficiently as the original polynomial.

▶ **Lemma 3.7** (Reversal Lemma). *Let* $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^i P_i(\mathbf{x})$ *be a polynomial computed by a formula (circuit)* Γ*, where* $P_r(\mathbf{x}) \cdot P_0(\mathbf{x}) \not\equiv 0$*. Let* $\tilde{P}(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^{r-i} P_i(\mathbf{x})$ *be its reversal. There exists a formula (circuit)* Δ *computing* $\tilde{P}$ *such that* $|\Delta| = 8r^2 \cdot |\Gamma|$*.*

We now connect the reversal operation to divisibility and irreducibility of polynomials.

▶ **Lemma 3.8** (Divisibility with Reversals). *Let* $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^i P_i(\mathbf{x})$*, with* $P_r(\mathbf{x}) \cdot P_0(\mathbf{x}) \not\equiv 0$ *and* $f(\mathbf{x}, y) \equiv \sum_{i=0}^{k} y^i f_i(\mathbf{x})$*, with* $f_k(\mathbf{x}) \cdot f_0(\mathbf{x}) \not\equiv 0$*, be two polynomials. In addition, let*

$\tilde{P}(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^{r-i} P_i(\mathbf{x})$ *and* $\tilde{f}(\mathbf{x}, y) \equiv \sum_{i=0}^{k} y^{k-i} f_i(\mathbf{x})$ *be their reversals. Then, we have that*

$$f \mid P \iff \tilde{f} \mid \tilde{P}.$$

Since divisibility is preserved by taking reversals, we have the following corollary:

▶ **Corollary 3.9** (Irreducibility of Reversals)**.** *Let* $P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^i P_i(\mathbf{x})$*, with* $P_r(\mathbf{x}) \cdot P_0(\mathbf{x}) \not\equiv 0$*, be an irreducible polynomial in* $\mathbb{F}[\mathbf{x}, y]$*. In addition, let* $\tilde{P}(\mathbf{x}, y) \equiv \sum_{i=0}^{r} y^{r-i} P_i(\mathbf{x})$ *be its reversal. Then, we have that*

$$P \text{ is irreducible} \iff \tilde{P} \text{ is irreducible.}$$

Another useful property of reversals is that if two univariate polynomials do not share a common root, then their reversals will not share any root either. This gives us the following lemma:

▶ **Lemma 3.10.** *If* $f(x), g(x) \in \mathbb{F}[x]$ *do not share any common roots, then their reversals* $\tilde{f}(x), \tilde{g}(x)$ *do not share any roots either.*

4 Properly Splitting Variable Restrictions

In this section, we study properties of pairs of polynomials $f(\mathbf{x}, y), g(\mathbf{x}, y)$ which share no common factor involving the variable y. We state a lemma on restrictions of the $\mathbf{x}$ variables of f and g that preserve the property that their restrictions share no common factors in y. We denote such restrictions as *properly splitting* variable restrictions.

▶ **Definition 4.1** (Properly Splitting Restrictions)**.** Let $\mathbf{x} = (x_1, \ldots, x_n)$, where $n \geq 1$, and let $f(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be an irreducible polynomial such that $\deg_y(f) \geq 1$. In addition, let $g(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial with $\deg_y(g) \geq 1$ that is not divisible by $f(\mathbf{x}, y)$. We say that $\mathbf{c} \in \mathbb{F}^n$ *properly splits* $f(\mathbf{x}, y)$ with respect to $g(\mathbf{x}, y)$ if the following conditions hold:

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

With the definition above, we are now ready to state the main lemma of this section. This lemma tells us that the set of restrictions that properly split an irreducible polynomial $f(\mathbf{x}, y)$ with respect to a polynomial $g(\mathbf{x}, y)$ that is not divisible by $f(\mathbf{x}, y)$ is the complement of an algebraic set. This implies that a random restriction of the variables $\mathbf{x}$ will properly split $f(\mathbf{x}, y)$ with respect to $g(\mathbf{x}, y)$.

▶ **Lemma 4.2.** *Let* $\mathbf{x} = (x_1, \ldots, x_n)$*, where* $n \geq 1$ *and* $f(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *be an irreducible polynomial such that* $\deg_y(f) \geq 1$*. In addition, let* $g(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *be a polynomial with* $\deg_y(g) \geq 1$ *that is not divisible by* $f(\mathbf{x}, y)$*. Then, there exists a nonzero polynomial* $G(\mathbf{x})$ *with* $\deg(G) \leq 2\deg(f)^2 + 2\deg(f)\deg(g)$ *for which the following holds: for any value* $\mathbf{c} \in \mathbb{F}^n$ *such that* $G(\mathbf{c}) \neq 0$*, we have that* $\mathbf{c}$ *properly splits* $f(\mathbf{x}, y)$ *with respect to* $g(\mathbf{x}, y)$*.*

5 Standard Forms and Approximation Polynomials

In this section we define the notion of standard forms in $\mathbb{F}(\mathbf{x})[y]$, that is, the ring of polynomials on the variable y with coefficients being rational functions on the variables $\mathbf{x}$.

We also define the approximation polynomial of a standard form. These concepts will be useful when factoring a polynomial $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$, since our factorization procedure will use standard forms to obtain the factors of $P(\mathbf{x}, y)$ that depend of the variable y. We begin with the following definition:

▶ **Definition 5.1** (Standard Form and Approximation Polynomials)**.** We say that $f(\mathbf{x}, y) \in \mathbb{F}(\mathbf{x})[y]$ is in *standard form* if

$$f(\mathbf{x}, y) \equiv \frac{1}{1 - g(\mathbf{x})} \cdot \sum_{i=0}^{k} f_i(\mathbf{x}) y^i,$$

where $f_i(\mathbf{x}), g(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$, $f_k(\mathbf{x}) \not\equiv 0$ and $g(\mathbf{0}) = 0$. Moreover, we will say that f is in *monic standard form* if $f_k(\mathbf{x}) \equiv 1 - g(\mathbf{x})$. For a given parameter $m \in \mathbb{N}$, we define the *approximation polynomial* of the standard form f to degree m, as the polynomial $\psi_{f,m}(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ given by

$$\psi_{f,m}(\mathbf{x}, \mathbf{y}) = (1 + g(\mathbf{x}) + \ldots + g(\mathbf{x})^m) \cdot \sum_{i=0}^{k} f_i(\mathbf{x}) y^i.$$

In order to state some useful properties of approximation polynomials, we will need to extend the definition of reversals to standard forms.

▶ **Definition 5.2.** Let $f(\mathbf{x}, y)$ be a standard form as above, with the additional condition that $f_0(\mathbf{x}) \not\equiv 0$. We define the reversal of $f(\mathbf{x}, y)$ as the following standard form:

$$\tilde{f}(\mathbf{x}, y) \equiv \frac{1}{1 - g(\mathbf{x})} \cdot \sum_{i=0}^{k} f_i(\mathbf{x}) y^{k-i}.$$

The following observations about standard forms reveal much of its usefulness when factoring a polynomial.

▶ **Observation 5.3.** *If $f(\mathbf{x}, y) \in \mathbb{F}(\mathbf{x})[y]$ is in standard form as above, notice that the following holds for all $m \in \mathbb{N}$:*

1. $H^{\mathbf{x}}_{\leq m}[(1 - g(\mathbf{x})) \cdot \psi_{f,m}(\mathbf{x}, y)] \equiv H^{\mathbf{x}}_{\leq m}[(1 - g(\mathbf{x})) \cdot f(\mathbf{x}, y)]$.
2. *If $m \geq \deg((1 - g(\mathbf{x})) \cdot f(\mathbf{x}, y))$, we have:*

$$H^{\mathbf{x}}_{\leq m}[(1 - g(\mathbf{x})) \cdot \psi_{f,m}(\mathbf{x}, y)] \equiv (1 - g(\mathbf{x})) \cdot f(\mathbf{x}, y).$$

3. $H^{\mathbf{x}}_{\leq m}[\psi_{\tilde{f},m}(\mathbf{x}, y)] \equiv H^{\mathbf{x}}_{\leq m}\left[\widetilde{\psi}_{f,m}(\mathbf{x}, y)\right]$.
4. *If $h(\mathbf{x}, y) \equiv f(\mathbf{x}, y + \gamma)$, where $\gamma \in \mathbb{F}$, we have that $h(\mathbf{x}, y)$ is also a standard form and*

$$H^{\mathbf{x}}_{\leq m}[\psi_{f,m}(\mathbf{x}, y + \gamma)] \equiv H^{\mathbf{x}}_{\leq m}[\psi_{h,m}(\mathbf{x}, y)].$$

6 Approximating the Roots of a Polynomial

In this section, we proceed in a similar way as in [2] and find approximations of the roots of a polynomial $P(\mathbf{x}, y)$ up to degree t. That is, as we defined in the introduction, we find polynomials $q_t(\mathbf{x})$ such that $H^{\mathbf{x}}_{\leq t}[P(\mathbf{x}, q_t(\mathbf{x}))] \equiv 0$. Moreover, we observe that under certain conditions on the polynomial $P(\mathbf{x}, y)$ these roots are well-defined and unique given their constant coefficient. This uniqueness condition will be useful because it will allow us to construct any factor of $P(\mathbf{x}, y)$ through the lifting procedure, since a factor $f(\mathbf{x}, y)$ of $P(\mathbf{x}, y)$ will share some of the roots of $P(\mathbf{x}, y)$. We begin with the approximation lemma:

▶ **Lemma 6.1** (Approximation Lemma). *Let* $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$, $P'(\mathbf{x}, y) \equiv \frac{\partial P}{\partial y}(\mathbf{x}, y)$ *and* $\mu \in \mathbb{F}$ *be such that* $P(\mathbf{0}, \mu) = 0$ *but* $P'(\mathbf{0}, \mu) = \xi \neq 0$. *Then, for each* $t \geq 0$, *there exists a* **unique** *polynomial* $q_t(\mathbf{x})$ *s.t.* $\deg(q_t) \leq t$, $q_t(\mathbf{0}) = \mu$ *and*

$$H^{\mathbf{x}}_{\leq t}[P(\mathbf{x}, q_t(\mathbf{x}))] \equiv 0.$$

Moreover, if P *can be computed by a formula (circuit)* Γ *such that its output gate is an addition gate, there is a formula (circuit)* Φ_t *for the polynomial* $q_t(\mathbf{x})$ *such that the output gate of* Φ_t *is an addition gate,* $\text{depth}(\Phi_t) \leq \text{depth}(\Gamma) + 2$ *and*

$$|\Phi_t| \leq 200(tr)^2 \binom{t + r + 1}{r + 1} \cdot |\Gamma|.$$

If we require the fanin of the formula (circuit) to be 2, *then the size of* Φ_t *does not change, and* $\text{depth}(\Phi_t) \leq \text{depth}(\Gamma) + 5r \log(t)$.

Now that we know that any root of a polynomial $P(\mathbf{x}, y)$ of small individual degree computed by a small formula can be approximated by a small formula, the next corollary uses the uniqueness of the approximation of the root to show that the same is true for any factor of $P(\mathbf{x}, y)$.

▶ **Corollary 6.2.** *Let* $P(\mathbf{x}, y)$ *and* $\mu \in \mathbb{F}$ *be defined as in Lemma 6.1 and for each* $t \in \mathbb{N}_0$, *let* $q_t(\mathbf{x})$ *be the unique polynomial obtained from Lemma 6.1. If* $h(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *is such that* $h(\mathbf{0}, \mu) = 0$, $\frac{\partial h}{\partial y}(\mathbf{0}, \mu) \neq 0$ *and there exist* $t \in \mathbb{N}$ *and* $Q(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *such that*

$$H^{\mathbf{x}}_{\leq t}[P(\mathbf{x}, y)] \equiv H^{\mathbf{x}}_{\leq t}[h(\mathbf{x}, y) \cdot Q(\mathbf{x}, y)], \tag{1}$$

then the polynomial $q_t(\mathbf{x})$ *also satisfies*

$$H^{\mathbf{x}}_{\leq t}[h(\mathbf{x}, q_t(\mathbf{x}))] \equiv 0, \ \forall t \geq 0.$$

7 Proof of the Main Theorem

In this section, we give the proof of our main theorem. In addition, we state the consequences of the main theorem for both small formula size and depth of circuits computing factors of polynomials with small bounded degree.

▶ **Lemma 7.1** (Main Lemma). *Let* $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ *be such that* $\deg_y(P) = r$, *and also* $\deg_{x_i}(P) \leq r, \forall i \in \{1, \ldots, n\}$. *Let* $P'(\mathbf{x}, y) \equiv \dfrac{\partial P}{\partial y}(\mathbf{x}, y)$. *In addition, let* $f(\mathbf{x}, y) \in \mathbb{F}(\mathbf{x})[y]$ *be in monic standard form and assume it is irreducible over* $\mathbb{F}(\mathbf{x})[y]$, *satisfying the following conditions:*

1. $f(\mathbf{x}, y) \mid P(\mathbf{x}, y)$[1],
2. $f(\mathbf{0}, y)$ *has exactly* $\deg_y(f)$ *distinct roots*[2],
3. $P'(\mathbf{0}, y)$ *and* $f(\mathbf{0}, y)$ *share no common roots.*

If there exists a formula (circuit) Γ *computing* P *with output gate being an addition gate,* $|\Gamma| = s$ *and* $\text{depth}(\Gamma) = d$, *then for every* $m \geq 1$, *there exist formulas (circuits)* Ψ_m *and* $\widetilde{\Psi}_m$ *with each output gate being a multiplication gate, of size*

$$\max(|\Psi_m|, |\widetilde{\Psi}_m|) \leq 300m^2r^3 \cdot \binom{m + r + 1}{r + 1} \cdot s$$

[1] Since $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$, this condition is equivalent to the existence of $Q(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ such that $f(\mathbf{x}, y) \cdot Q(\mathbf{x}, y) \equiv P(\mathbf{x}, y)$.

[2] Note that we can evaluate $f(\mathbf{x}, y)$ at $\mathbf{x} = \mathbf{0}$, since $f(\mathbf{x}, y)$ is in standard form.

and depth $\max(\text{depth}(\Psi_m), \text{depth}(\widetilde{\Psi}_m)) \leq d+3$ *such that*

$$H^{\mathbf{x}}_{\leq m}[\Psi_m] \equiv H^{\mathbf{x}}_{\leq m}[\psi_{f,m}(\mathbf{x}, y)] \quad \textit{and}$$

$$H^{\mathbf{x}}_{\leq m}[\widetilde{\Psi}_m] \equiv H^{\mathbf{x}}_{\leq m}[\psi_{\tilde{f},m}(\mathbf{x}, y)].$$

If we require the in-degree of the formula (circuit) to be 2*, then the size of* Ψ_m *or* $\widetilde{\Psi}_m$ *does not change, and* $\max(\text{depth}(\Psi_m), \text{depth}(\widetilde{\Psi}_m)) \leq d + 10r \log m$.

With the Main Lemma stated above, we are now able to state and prove our main theorem.

▶ **Theorem 7.2** (Main Theorem). *Let* $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}] \setminus \{0\}$ *be such that* $\deg_{x_i}(P) \leq r$, $1 \leq i \leq n$, $P(\mathbf{0}) \neq 0$ *and let* Γ *be a formula (circuit) of size* s *and depth* d *computing* P*. Let* $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *be a factor of* $P(\mathbf{x})$*, and let* m *be a positive integer. There exists a polynomial* $G(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *of total degree* $\deg(G) \leq 4r^3n^3$ *such that if* $\mathbf{c} \in \mathbb{F}^n$ *satisfies* $G(\mathbf{c}) \neq 0$ *then there exists a formula* Φ_m *whose output gate is a multiplication gate and for which*

$$\text{depth}(\Phi_m) \leq d + 4^3,$$

$$|\Phi_m| \leq 60000m^2r^8n \cdot \binom{m+r+1}{r+1} s \quad \textit{and}$$

$$H^{\mathbf{x}}_{\leq m}[\Phi_m(\mathbf{x})] \equiv H^{\mathbf{x}}_{\leq m}[f(\mathbf{x}+\mathbf{c})].$$

If we require the in-degree of the formula (circuit) to be 2*, then the size of* Φ_m *does not change, and* $\text{depth}(\Phi_m) \leq d + 20r \log m$.

Proof. The proof of the theorem is by induction on the number of variables. The bound is trivial in the univariate case, since if $f(x), P(x) \in \mathbb{F}[x]$, where $\deg(f) = k \leq r$ and $f \mid P$, then we can write

$$f(x) = c \cdot \prod_{i=1}^{k} (x - \mu_i),$$

which can be trivially computed by a formula Ψ of size $\leq 50k$ and depth 2. In this case, setting $G(x)$ to be any constant polynomial, for instance $G(x) \equiv 1$, $\mathbf{c} = \mathbf{0}$ and $\Phi_m = \Psi$, takes care of the base case.

Hence, let's assume that the claim is true for polynomials $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}] = \mathbb{F}[x_1, \ldots, x_n]$ with $P(\mathbf{0}) \neq 0$, for some $n \geq 1$. Now we will prove that the same bounds hold for polynomials $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ s.t. $P(\mathbf{0}, 0) \neq 0$. Let $P(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a polynomial computed by Γ and $f(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ be a factor of $P(\mathbf{x}, y)$. We can assume that $f(\mathbf{x}, y)$ and $P(\mathbf{x}, y)$ depend on y, otherwise we can simply restrict the formula Γ to $\Gamma|_{y=0}$, and by the induction hypothesis the result follows.

Let

$$P(\mathbf{x}, y) \equiv \sum_{i=0}^{r} C_i(\mathbf{x}) y^i, \quad \text{and} \quad f(\mathbf{x}, y) \equiv q(\mathbf{x}) \cdot \prod_{i=1}^{t} f_i(\mathbf{x}, y)^{e_i}, \quad \text{with}$$

$$f_i(\mathbf{x}, y) \equiv \sum_{j=0}^{k_i} f_{ij}(\mathbf{x}) y^j, \text{ where } f_{i0}(\mathbf{x}) \cdot f_{ik_i}(\mathbf{x}) \not\equiv 0, \quad \forall 1 \leq i \leq t.$$

[3] If the bottom gates are addition gates, then the depth is bounded by $d + 3$.

where each $f_i(\mathbf{x}, y) \in \mathbb{F}[\mathbf{x}, y]$ is an irreducible polynomial. Since $P(\mathbf{0}, 0) \neq 0$, we have that $C_0(\mathbf{x}) \equiv P(\mathbf{x}, 0) \not\equiv 0$, and moreover, that $C_0(\mathbf{0}) \neq 0$. Let

$$u(\mathbf{x}) \equiv f(\mathbf{x}, 0) \equiv q(\mathbf{x}) \cdot \prod_{i=1}^{t} f_{i0}(\mathbf{x})^{e_i}.$$

Notice that $f(\mathbf{x}, y) \mid P(\mathbf{x}, y) \Rightarrow u(\mathbf{x}) \mid C_0(\mathbf{x})$. In addition, notice that $C_0(\mathbf{0}) \neq 0$ and $C_0(\mathbf{x})$ can be computed by the formula $\Gamma|_{y=0}$, where $|\Gamma|_{y=0}| \leq |\Gamma|$ and $\text{depth}(\Gamma|_{y=0}) \leq \text{depth}(\Gamma)$. Therefore, by induction hypothesis, there exists $H(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ with $\deg(H) \leq 4r^3n^3$ such that for any $\mathbf{a} \in \mathbb{F}^n$ for which $H(\mathbf{a}) \neq 0$, there exists a formula Λ_m with output gate being a multiplication gate, such that

$$\text{depth}(\Lambda_m) \leq d + 4,$$

$$|\Lambda_m| \leq 60000m^2r^8n \cdot \binom{m + r + 1}{r + 1} s \quad \text{and}$$

$$H^{\mathbf{x}}_{\leq m}[\Lambda_m(\mathbf{x})] \equiv H^{\mathbf{x}}_{\leq m}[u(\mathbf{x} + \mathbf{a})].$$

Now that we have an approximation to the factor $u(\mathbf{x})$, which is the constant term of the polynomial $f(\mathbf{x}, y)$ when seen as a polynomial in the variable y, we want to use Lemma 7.1 to find the factors of $f(\mathbf{x}, y)$ that contain y. For this, we will first need to find polynomials $D_i(\mathbf{x}, y)$ with small formulas such that $f_i(\mathbf{x}, y) \mid D_i(\mathbf{x}, y)$ and each D_i is square-free with respect to $f_i(\mathbf{x}, y)$.

Fortunately, Lemma 3.5 tells us that for each (irreducible) polynomial $f_i(\mathbf{x}, y)$, we can find formulas Δ_i of size $\leq 9r^2|\Gamma|$ computing polynomials $D_i(\mathbf{x}, y)$ such that $\deg_{x_j}(D_i) \leq r,\ 1 \leq j \leq n,\ \deg_y(D_i) \leq r,\ f_i(\mathbf{x}, y) \mid D_i(\mathbf{x}, y)$ but $f_i(\mathbf{x}, y) \nmid \dfrac{\partial D_i}{\partial y}(\mathbf{x}, y)$. Moreover these formulas have an addition gate as output gate.

Since $f_i(\mathbf{x}, y)$ is irreducible with $\deg_y(f_i) \geq 1$ and $f_i(\mathbf{x}, y) \nmid \dfrac{\partial D_i}{\partial y}(\mathbf{x}, y)$, Lemma 4.2 implies that there exists a polynomial $G_i(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ with

$$\deg(G_i) \leq 2\deg(f_i)^2 + 2\deg(f_i)\deg(D_i) \leq 4r^2n^2$$

such that for any $\mathbf{c} \in \mathbb{F}^n$ where $G_i(\mathbf{c}) \neq 0$ we have that $\mathbf{c}$ properly splits $f_i(\mathbf{c}, y)$ with respect to $\dfrac{\partial D_i}{\partial y}(\mathbf{c}, y)$.

Let

$$G(\mathbf{x}, y) \equiv H(\mathbf{x}) \cdot C_0(\mathbf{x}) \cdot \prod_{i=1}^{t} G_i(\mathbf{x}) \ \text{ and } \ (\mathbf{c}, \gamma) \in \mathbb{F}^{n+1} \ \text{ be s.t. } \ G(\mathbf{c}, \gamma) \neq 0.^4$$

[4] At first, it may seem strange that $G(\mathbf{x}, y)$ does not depend on the variable y, since if we continued this argument by induction we would arrive at the conclusion that $G(\mathbf{x}, y)$ is the constant polynomial. However, notice that even though $H(\mathbf{x})$ does not depend on the variable x_n, the polynomial $G(\mathbf{x}, y)$ depends on x_n, since the polynomials $C_0(\mathbf{x})$ and $G_i(\mathbf{x})$ depend on x_n. The right way to see this dependence is the following: $G(\mathbf{x}, y)$ depends on every variable except the variable used by the lifting procedure, which in this case is the variable y. Hence, we will have that $H(\mathbf{x})$ depends on all the variables except x_n (if we choose to perform the lifting with respect to x_n).

Denote

$$Q(\mathbf{x},y) \equiv P(\mathbf{x}+\mathbf{c},y) \equiv \sum_{i=0}^{r} Q_i(\mathbf{x})y^i, \qquad h_i(\mathbf{x},y) \equiv f_i(\mathbf{x}+\mathbf{c},y) \equiv \sum_{j=0}^{k_i} h_{ij}(\mathbf{x})y^j \text{ and}$$

$$h(\mathbf{x},y) \equiv f(\mathbf{x}+\mathbf{c},y) \equiv q(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} h_i(\mathbf{x},y)^{e_i}.$$

Since $h_{i0}(\mathbf{x}) \equiv f_{i0}(\mathbf{x}+\mathbf{c},0) \mid P(\mathbf{x}+\mathbf{c},0) \equiv C_0(\mathbf{x}+\mathbf{c})$ and $C_0(\mathbf{c}) \neq 0$ (because $G(\mathbf{c},\gamma) \neq 0$), we have that $h_{i0}(\mathbf{0}) \neq 0$, for all $1 \leq i \leq t$. Hence, after normalization by a proper field element, we can write each h_{i0} in the following form:

$$h_{i0}(\mathbf{x}) = 1 - g_i(\mathbf{x}), \quad \text{where } g_i(\mathbf{0}) \equiv 0.$$

In addition, notice that $f_{ik_i}(\mathbf{x}) \not\equiv 0 \Rightarrow h_{ik_i}(\mathbf{x}) \equiv f_{ik_i}(\mathbf{x}+\mathbf{c}) \not\equiv 0$.

Moreover, notice that $f_i(\mathbf{x},y)$ is irreducible with $f_{i0}(\mathbf{x}) \cdot f_{ik_i}(\mathbf{x}) \not\equiv 0$ implies that $h_i(\mathbf{x},y)$ is irreducible with $h_{i0}(\mathbf{x}) \cdot h_{ik_i}(\mathbf{x}) \not\equiv 0$, which implies (by Corollary 3.9) that the polynomial $\widetilde{h_i}(\mathbf{x},y) \equiv \sum_{j=0}^{k_i} h_{ij}(\mathbf{x})y^{k_i-j}$ is irreducible in $\mathbb{F}[\mathbf{x},y]$. Hence, we have that $\ell_i(\mathbf{x},y) \equiv \dfrac{\widetilde{h_i}(\mathbf{x},y)}{h_{i0}(\mathbf{x})}$ is a monic irreducible standard form in $\mathbb{F}(\mathbf{x})[y]$.

Because $f_i(\mathbf{x},y) \mid D_i(\mathbf{x},y)$ and $f_i(\mathbf{x},y) \nmid \dfrac{\partial D_i}{\partial y}(\mathbf{x},y)$, by Lemma 3.8 we obtain that $h_i(\mathbf{x},y) \mid E_i(\mathbf{x},y) \equiv D_i(\mathbf{x}+\mathbf{c},y)$ and $h_i(\mathbf{x},y) \nmid \dfrac{\partial E_i}{\partial y}(\mathbf{x},y) \equiv \dfrac{\partial D_i}{\partial y}(\mathbf{x}+\mathbf{c},y)$.

Since $h_i(\mathbf{0},y) \equiv f_i(\mathbf{c},y)$, we also have that $h_i(\mathbf{0},y)$ has no common roots with $\dfrac{\partial E_i}{\partial y}(\mathbf{0},y)$. The following claim shows that $\ell_i(\mathbf{x},y)$ satisfies the conditions of Lemma 7.1.

▶ **Claim 7.3.** *For each $i \in \{1,\ldots,t\}$, the monic irreducible standard form $\ell_i(\mathbf{x},y) \equiv \dfrac{\widetilde{h_i}(\mathbf{x},y)}{h_{i0}(\mathbf{x})}$ and the polynomial $\tilde{E}_i(\mathbf{x},y)$ satisfy the conditions of Lemma 7.1.*

Proof of claim. Notice that conditions (i) and (ii) from Lemma 7.1 follow from the fact that $h_i(\mathbf{x},y) \mid E_i(\mathbf{x},y)$ and Lemmas 3.8 and 4.2. Condition (iii) follows from the fact that $\dfrac{h_i(\mathbf{0},y)}{h_{i0}(\mathbf{0})} \equiv h_i(\mathbf{0},y)$ shares no common roots with $\dfrac{\partial E_i}{\partial y}(\mathbf{0},y)$ and from Lemma 3.10.

This finishes the proof of the claim. ◀

Now that we have rational functions in monic standard form that are, in a certain sense, computing the reversal of each $f_i(\mathbf{x},y)$, we can use the main lemma to lift the factorization of the approximation polynomial of $f_i(\mathbf{x},y)/f_{i0}(\mathbf{x})$.[5]

Since each $\ell_i(\mathbf{x},y)$ and $\tilde{E}_i(\mathbf{x},y)$ satisfy the conditions of Lemma 7.1, and $\tilde{E}_i(\mathbf{x},y)$ can be computed by a formula Υ_i of size $|\Upsilon_i| \leq 180r^4 \cdot |\Gamma| = 180r^4 s$ and depth $\text{depth}(\Upsilon_i) \leq d+1$ (since Υ_i is a shift of $\widetilde{\Delta}_i$), we have that there exists a formula $\Psi_{i,m}$ having as output gate a multiplication gate, $\text{depth}(\Psi_{i,m}) \leq \text{depth}(\Upsilon_i) + 3 \leq d+4$ and size

$$|\Psi_{i,m}| \leq 300m^2r^3 \cdot \binom{m+r+1}{r+1} \cdot 180r^4 \cdot s \leq 60000m^2r^7 \cdot \binom{m+r+1}{r+1} \cdot s$$

[5] In actuality, we are performing a lift of a shift of $f_i(\mathbf{x},y)$.

such that

$$H^{\mathbf{x}}_{\leq m}[\Psi_{i,m}] \equiv H^{\mathbf{x}}_{\leq m}[\psi_{\widetilde{\ell_i(\mathbf{x},y)},m}(\mathbf{x},y)].$$

By Observation 5.3, we have that

$$H^{\mathbf{x}}_{\leq m}[h_{i0}(\mathbf{x}) \cdot \psi_{\widetilde{\ell_i(\mathbf{x},y)},m}(\mathbf{x},y)] \equiv H^{\mathbf{x}}_{\leq m}[\widetilde{\ell_i}(\mathbf{x},y) \cdot h_{i0}(\mathbf{x})] \equiv H^{\mathbf{x}}_{\leq m}[h_i(\mathbf{x},y)], \quad \text{and also}$$

$$H^{\mathbf{x}}_{\leq m}[h_{i0}(\mathbf{x}) \cdot \psi_{\widetilde{\ell_i(\mathbf{x},y)},m}(\mathbf{x},y+\gamma)] \equiv H^{\mathbf{x}}_{\leq m}[h_i(\mathbf{x},y+\gamma)].$$

In addition, from the formulas $\Psi_{i,m}$ and from the fact that $\sum_{i=1}^{t} e_i \leq r$, we have that the formula given by $\Psi_m = \prod_{i=1}^{t} \Psi_{i,m}^{e_i}$ is of size

$$|\Psi_m| \leq \sum_{i=1}^{t} e_i \cdot |\Psi_{i,m}| \leq r \cdot \max_{1\leq i\leq t}(|\Psi_{i,m}|) \leq 60000m^2r^8 \cdot \binom{m+r+1}{r+1} \cdot s$$

and computes the following polynomial:

$$H^{\mathbf{x}}_{\leq m}[\Psi_m(\mathbf{x},y)] \equiv H^{\mathbf{x}}_{\leq m}\left[\prod_{i=1}^{t} \psi_{\widetilde{\ell_i(\mathbf{x},y)},m}(\mathbf{x},y+\gamma)^{e_i}\right].$$

Now that we found a formula computing the approximation polynomials $\psi_{\widetilde{\ell_i(\mathbf{x},y)},m}(\mathbf{x},y+\gamma)$, we can multiply them by $h_{i0}(\mathbf{x},y)$ and via Observation 5.3 obtain the polynomials $h_i(\mathbf{x},y)$, which are the shifts of $f_i(\mathbf{x},y)$. Since Ψ_m computes all of the approximation polynomials, and Λ_m computes all of the leading coefficients, by combining them we can recover the factor $f(\mathbf{x},y)$. This is what we do next.

Multiplying Ψ_m by Λ_m, we have that the formula $\Phi_m = \Lambda_m \cdot \Psi_m$ is such that

$$|\Phi_m| \leq |\Lambda_m| + |\Psi_m| \leq 60000m^2r^8(n+1) \cdot \binom{m+r+1}{r+1} \cdot s$$

and

$$\begin{aligned}
H^{\mathbf{x}}_{\leq m}[\Phi_m(\mathbf{x},y)] &\equiv H^{\mathbf{x}}_{\leq m}[\Lambda_m \cdot \Psi_m] \equiv H^{\mathbf{x}}_{\leq m}\left[u(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} \psi_{\frac{h_i(\mathbf{x},y)}{h_{i0}(\mathbf{x})},m}(\mathbf{x},y+\gamma)^{e_i}\right] \\
&\equiv H^{\mathbf{x}}_{\leq m}\left[q(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} f_{i0}(\mathbf{x}+\mathbf{c})^{e_i} \cdot \prod_{i=1}^{t} \psi_{\frac{h_i(\mathbf{x},y)}{h_{i0}(\mathbf{x})},m}(\mathbf{x},y+\gamma)^{e_i}\right] \\
&\equiv H^{\mathbf{x}}_{\leq m}\left[q(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} \left(h_{i0}(\mathbf{x}) \cdot \psi_{\frac{h_i(\mathbf{x},y)}{h_{i0}(\mathbf{x})},m}(\mathbf{x},y+\gamma)\right)^{e_i}\right] \\
&\equiv H^{\mathbf{x}}_{\leq m}\left[q(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} h_i(\mathbf{x},y+\gamma)^{e_i}\right] \\
&\equiv H^{\mathbf{x}}_{\leq m}\left[q(\mathbf{x}+\mathbf{c}) \cdot \prod_{i=1}^{t} f_i(\mathbf{x}+\mathbf{c},y+\gamma)^{e_i}\right] \equiv H^{\mathbf{x}}_{\leq m}[f(\mathbf{x}+\mathbf{c},y+\gamma)].
\end{aligned}$$

Since

$$\deg(G(\mathbf{x},y)) \leq \deg(H) + \deg(C_0) + \sum_{i=1}^{t} \deg(G_i) \leq 4r^3n^3 + rn + r \cdot 4r^2n^2 \leq 4r^3(n+1)^3,$$

this finishes the induction, and therefore the proof of the theorem. It is clear from the proof, via Observation 3.6, that if we restrict the in-degree to 2, we obtain the desired bound on the depth. ◀

As a corollary of the main theorem, we obtain:

▶ **Corollary 7.4** (Small Formula – Restatement of Theorem 1.2)**.** *Let $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}] \setminus \{0\}$ be such that $\deg_{x_i}(P) \leq r$, $1 \leq i \leq n$, and let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a factor of P. If there exists a formula Γ of size s and depth d computing P, then there exists a formula Φ of depth $\text{depth}(\Phi) \leq d+5$ and size*

$$|\Phi| = O\left(n^3 r^{12} \cdot \binom{nr+r+1}{r+1} s\right) = \textit{poly}((nr)^r, s)$$

such that

$$\Phi(\mathbf{x}) \equiv f(\mathbf{x}).$$

If we require the in-degree of the formula (circuit) to be 2, then the size of Φ does not change, and $\text{depth}(\Phi) \leq d + 30r\log(nr)$.

Proof. Let $\mathbf{c} \in \mathbb{F}^n$ be such that $P(\mathbf{c}) \neq 0$. Such a $\mathbf{c}$ exists since $P(\mathbf{x})$ is nonzero. This implies that $Q(\mathbf{x}) \equiv P(\mathbf{x}+\mathbf{c})$ is computed by the formula $\Delta(\mathbf{x}) = \Gamma(\mathbf{x}+\mathbf{c})$, of size $\leq 2|\Gamma| = 2s$, $\text{depth}(\Delta) \leq d+1$ and is such that $Q(\mathbf{0}) = P(\mathbf{c}) \neq 0$. Hence, by Theorem 7.2, we have that there exists polynomial $G(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ of degree $\deg(G) \leq 4r^3n^3$ such that for any $\mathbf{a} \in \mathbb{F}^n$ for which $G(\mathbf{a}) \neq 0$, there is a formula Φ_{nr} whose output gate is a multiplication gate for which $\text{depth}(\Phi_{nr}) \leq \text{depth}(\Delta) + 3 \leq d+4$, of size

$$|\Phi_{nr}| \leq 120000(nr)^2 r^8 n \cdot \binom{nr+r+1}{r+1} s \quad \text{and such that}$$

$$H^{\mathbf{x}}_{\leq nr}[\Phi_{nr}(\mathbf{x})] \equiv H^{\mathbf{x}}_{\leq nr}[f(\mathbf{x}+\mathbf{c}+\mathbf{a})] \equiv f(\mathbf{x}+\mathbf{c}+\mathbf{a}), \quad \text{since } nr \geq \deg(f).$$

By the interpolation Lemma 3.4, we obtain that there exists a formula Φ' of size

$$|\Phi'| \leq 9r^2 \cdot |\Phi_{nr}|$$

and $\text{depth}(\Phi') \leq d+5$ such that $\Phi'(\mathbf{x}) \equiv f(\mathbf{x}+\mathbf{c}+\mathbf{a})$. By shifting the inputs of the formula Φ' by $-(\mathbf{a}+\mathbf{c})$, we have that the new formula just obtained, call it Φ, computes the polynomial $f(\mathbf{x})$, as we wanted. It is easy to see that Φ has the desired upper bound on its size. It is also clear from the proof that if we restrict the in-degree of the formulas (circuits) to be 2, we obtain the desired bounds on the depth. This finishes the proof. ◀

8 Conclusion

Besides solving a question posed by Kopparty et al. [10] and Open Problem 19 in [15] for the class of bounded individual degree polynomials, notice that Lemma 7.1 and Theorem 7.2 also provide a framework to convert formulas (circuits) for the approximate roots of a polynomial into actual formulas (circuits) for factors of the same polynomial. Since Lemma 7.1, and therefore Theorem 7.2, uses the Approximation Lemma (Lemma 6.1) as a black-box, any improvements on Lemma 6.1 would lead to better bounds on the size of the formulas for the factors of the input polynomial. Hence, if one can remove the exponential dependence on the parameter r (the bound on the individual degrees) in the Approximation Lemma, one can fully solve the questions above. This is the main open question left by this work.

Acknowledgements. The author would like to thank his advisor Zeev Dvir for all the helpful discussions and encouragement throughout the course of this work.

References

1 Benny Chor and Ronald L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5):901–909, 1988.

2 Z. Dvir, A. Shpilka, and A. Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. on Computing*, 39(4):1279–1293, 2009.

3 J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.

4 J. Von Zur Gathen and E. Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985.

5 V. Guruswami and M. Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Trans. Inf. Theor.*, 45(6):1757–1767, September 2006.

6 V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

7 E. Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. on computing*, 14(2):469–489, 1985.

8 E. Kaltofen. Factorization of polynomials given by straight-line programs. In S. Micali, editor, *Randomness in Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press, 1989.

9 E. Kaltofen. Polynomial factorization: a success story. In *ISSAC*, pages 3–4, 2003.

10 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 169–180, 2014.

11 A. K. Lenstra, H. W. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982.

12 Hendrik W Lenstra Jr. Finding small degree factors of lacunary polynomials. *Number theory in progress*, 1:267–276, 1999.

13 R. Oliveira. Factors of low individual degree polynomials. `http://www.cs.princeton.edu/~rmo/papers/small-depth-factors.pdf`, 2015.

14 A. Shpilka and I. Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *ICALP (1)*, pages 408–419, 2010.

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

16 Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.