

Extracting Roots of Arithmetic Circuits

by Adapting Numerical Methods

Maurice Jansen*

School of Informatics, The University of Edinburgh

maurice.julien.jansen@gmail.com

Abstract: For two polynomials $f \in \mathbb{F}[x_1, x_2, \dots, x_n, y]$ and $p \in \mathbb{F}[x_1, x_2, \dots, x_n]$, we say that p is a *root* of f , if $f(x_1, x_2, \dots, x_n, p) \equiv 0$. We study the relation between the arithmetic circuit sizes of f and p for general circuits and algebraic branching programs. An algebraic branching program (ABP) is given by a layered directed acyclic graph with source σ and sink τ , whose edges are labeled by variables or field constants. It computes the sum of weights of all paths from σ to τ , where the weight of a path is defined as the product of edge-labels on the path. For the size of an ABP we count the number of nodes in the underlying graph.

We address the following fundamental question: suppose the polynomial f can be computed by an ABP of size s . Is the ABP size of every root p of f guaranteed to be bounded by a polynomial in s ? For general circuits it is known that the circuit size of any root p of a polynomial f with circuit size s is at most $\text{poly}(s, \text{deg}(p), m)$, where m is the *multiplicity* of p in f , i.e. m is the largest number such that $(p - y)^m$ divides f . This bound follows from a result about factors of arithmetic circuits independently obtained by Kaltofen [1] and Bürgisser [2].

In this paper, we study the above question for ABPs for the canonical case where f is assumed to factor as $f = p_0 \cdot (p_1 - y)(p_2 - y) \dots (p_r - y)$, for $p_0, p_1, \dots, p_r \in \mathbb{F}[x_1, x_2, \dots, x_n]$ with $p_0 \neq 0$, and where p_1, p_2, \dots, p_r are pairwise distinct, i.e. all multiplicities are one. Our main result is that for this situation, provided \mathbb{F} has characteristic zero, any root p_i can be computed by an ABP of size polynomial in s . This demonstrates an important special case where the answer to the above mentioned question is affirmative.

To prove the above result, we view the question as a problem of computing eigenvalues. Roughly, the p_i s are made to appear as the eigenvalues of some matrix over the field $\mathbb{F}(x_1, x_2, \dots, x_n)$ of rational functions. This problem is then solved by adapting the numerical method of power iteration to our situation. Using power iteration makes the computation amenable to be coded out as an ABP, since ABPs can efficiently compute iterated matrix multiplication.

In this work we adapt techniques which are well-known from numerical analysis, for use in the area of arithmetic circuit complexity. Staying with this theme, we also improve the above mentioned $\text{poly}(s, \text{deg}(p), m)$ bound for the circuit size of a root p of a polynomial f computed by an (unrestricted) arithmetic circuit of size s . Rather than applying [1, 2], we develop a discrete analogue of Newton's Method.

Keywords: arithmetic circuits, derandomization, polynomial identity testing, root extraction, numerical analysis.

1 Introduction

For informal use, let us say an arithmetic circuit class \mathcal{C} is *closed under taking roots*, if roots of (families of) polynomials in \mathcal{C} also belong to the class \mathcal{C} .

*This work was conducted while the author was a Postdoctoral Fellow at the Institute for Theoretical Computer Science of Tsinghua University, and it was supported in part by the National Natural Science Foundation of China Grant 60553001, 61073174, 61033001 and the National Basic Research Program of China Grant 2007CB807900, 2007CB807901.

Important consequences follow for classes that enjoy this property either completely, or for which a 'fairly decent' root extraction lemma can be proved. Most notably, such a lemma is a crucial tool for the conditional derandomization of *polynomial identity testing* (PIT) for the class \mathcal{C} . For the latter well-known problem one is given an arithmetic circuit Φ , and the problem is to decide whether the polynomial computed by Φ is identical to the zero polynomial or not. Due to a result independently obtained by Kaltofen [1] and Bürgisser [2], we know that the class VP of *poly*

degree polynomial families computable by *poly* size arithmetic circuits is even closed under taking *factors*, which implies the closure under taking roots. In their seminal paper on PIT, Kabanets and Impagliazzo [3] use this to give a deterministic subexponential time algorithm for identity testing ‘VP-circuits’, under the assumption that there exist some explicit polynomial f_n that requires super-polynomial arithmetic circuit size.

For more restricted classes \mathcal{C} , it is interesting to consider the question whether PIT for \mathcal{C} can be achieved deterministically under any weaker assumptions. When using the framework of [3], the situation where \mathcal{C} is closed under taking roots is ideal, since any loss incurred at the root extraction stage is directly reflected in the quality of the resulting hardness to randomness conversion. Examples of research efforts that follow this approach are the works by Dvir, Shpilka and Yehudayoff [4] and Jansen [5].

In [4] a root extraction lemma is proved for constant depth arithmetic circuits with $O(1)$ loss in the depth, that works well under the promise that the computed polynomials are of low degree. Consequently, a corresponding hardness to randomness conversion is obtained that applies to a low degree promise version of PIT for depth $d - O(1)$ circuits, assuming the existence of an explicit polynomial that is hard for arithmetic circuits of constant depth d . For ABPs a root extraction lemma is proved in [5], again with parameters working well only for low degree polynomials. Using this, it is proved that a certain low degree promise version of PIT for ABPs can be solved deterministically in subexponential time, assuming some explicit polynomial is hard for ABPs. In this paper we make progress towards showing that the arithmetic circuit class VDET of polynomial families computable by *poly* size ABPs is closed under taking roots. The latter statement, *if true*, would yield¹ a deterministic subexponential time PIT algorithm for VDET, under the assumption that there exists some explicit family of polynomials that requires ABPs of super-polynomial size.

Already implicit in [4, 5] was the use of a discrete analogue of Newton’s Method. We will revisit this, to give a self-contained proof of the fact that VP is closed under taking roots. The resulting argument

¹Already if the main result of this paper (Theorem 1) can be generalized to deal with arbitrary multiplicities, one would obtain this, based on the assumption that there exists an explicit family of polynomials that requires super-polynomial ABP size over *all* fields.

may serve as a conceptual simplification in [3], in the sense that calling upon the more involved works [1, 2] is avoided. For ABPs however, it is hard to imagine that this technique will ultimately lead to an optimal root extraction lemma. Therefore, in this paper we take an entirely new approach. We cast the problem as a task of computing eigenvalues, and adapt the method of power iteration to our domain. This way, since ABPs can efficiently compute matrix multiplication, we avoid the explosion in ABP size seemingly inherent to adaptations of Newton’s Method.

In the continuous domain, given a real $s \times s$ matrix M , say with real eigenvalues $\lambda_1 > \lambda_2 > \dots > \lambda_s > 0$ and a corresponding independent set of unit eigenvectors v_1, v_2, \dots, v_s , a well-known heuristic for finding an approximation to the largest eigenvalue λ_1 is to apply power iteration. Here, starting with some vector u that is typically selected at random, writing u in the eigenbasis as $u = a_1 v_1 + a_2 v_2 + \dots + a_s v_s$, for certain scalars a_i , one applies a large power of M to u to obtain $M^e u = a_1 \lambda_1^e v_1 + a_2 \lambda_2^e v_2 + \dots + a_s \lambda_s^e v_s$. After normalization, the term $a_1 \lambda_1^e v_1$ will be the dominant one, and thus the normalized sum will converge to v_1 as $e \rightarrow \infty$. Once an approximation \tilde{v}_1 to v_1 is obtained, one may approximate λ_1 by computing, for some nonzero component $(\tilde{v}_1)_\ell$, the ratio $(M\tilde{v}_1)_\ell / (\tilde{v}_1)_\ell$.

For the main development of this paper we will adapt the method of power iteration to construct small ABPs for roots of ABPs. Typically, in practice no good bounds are available for the rate of convergence of power iteration. It is worth mentioning that in our adaption we manage to avoid this, as well as several other crucial issues that arise along the way. We postpone a discussion of our techniques to Section 1.2. This work provides a case study of how standard tools from numerical analysis can be made available in the area of arithmetic circuit complexity, and hopefully stimulates further research into this direction.

1.1 Results

Our main result is the following theorem:

Theorem 1. *Let \mathbb{F} be a field of characteristic zero. Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n, y]$ be a nonzero polynomial that can be computed by an ABP of size s . Suppose f factors as $f = p_0(p_1 - y)(p_2 - y) \dots (p_r - y)$, where $\{p_0, p_1, p_2, \dots, p_r\} \subset \mathbb{F}[x_1, x_2, \dots, x_n]$ and p_1, p_2, \dots, p_r are pairwise distinct. Then every p_i has*

an ABP of size at most polynomial² in d, r and s , where $d = \max_{i \in [r], p_i \neq 0} \deg(p_i)$.

An ABP of size s computes a polynomial for which both its total degree and the individual degree of any variable is bounded by s . This implies that in the above theorem both r and d are at most s . For comparison, Lemma 2.10 of [5] yields³ an upper bound of $s \cdot 2^{O(\log^2 \deg(p_i))} r^{4+\log \deg(p_i)}$ for the size of an ABP for p_i .

For our second result, define the function $\mathcal{M}(d)$ to be an upper bound on the size of an arithmetic circuit for computing the multiplication of two univariate polynomial g and h in $\mathbb{F}[z]$ of degree at most d , given the coefficients of g and h as input variables. By a result of Cantor and Kaltofen [6], one can take $\mathcal{M}(d) = O(d \log d \log \log d)$, over *any* field \mathbb{F} . For (unrestricted) arithmetic circuits we have the following theorem:

Theorem 2. *Let \mathbb{F} be a field of characteristic zero. Let $f \in \mathbb{F}[x_1, x_2, \dots, x_n, y]$ be a polynomial of degree $r > 0$ that is computable by an arithmetic circuit of size s and let $p \in \mathbb{F}[X]$ be a nonconstant root of f for y , i.e. $f(x_1, x_2, \dots, x_n, p) \equiv 0$ and $p \notin \mathbb{F}$. Then p can be computed by an arithmetic circuit of size $O(\mathcal{M}(m)\mathcal{M}(\deg(p)) \cdot \deg(p) \cdot s)$, where m is the multiplicity of the root p in f .*

Due to a Lemma by Gauss (Lemma 1), in the above situation p is a root of f if and only $p - y$ is an irreducible factor of f in $\mathbb{F}[x_1, x_2, \dots, x_n, y]$. Using [1, 2] to obtain arithmetic circuits for the factor $p - y$, as done in [3], yields a circuit for the root p of size $O(\mathcal{M}(\deg(p))^3 m (s + \deg(p) \log m))$. It can be verified that our result is an improvement over the bound obtained this way.

1.2 Outline of the proof of Theorem 1

We roughly follow the following program:

1. Reduction to ‘nice’ polynomials.

The first step in the proof is to show that the gen-

²In this paper our aims are purely theoretical. The exponents of this polynomial are *large*, i.e. without making efforts to optimize a bound of $O(r^{2556} d^{84} s^{2160})$ can be given. To some extent it is remarkable that a polynomial bound can be given at all. If f satisfies a certain ‘niceness’ condition the bound improves somewhat down to $O(r^{396} d^{84} s^{180})$.

³Note that this lemma is stated for *skew circuits*, but inspection of the proof yields the given bound.

eral case of Theorem 1 reduces to the case where $p_0 = 1$ and the other p_i s are nonconstant polynomials, with the constant terms $p_1(0), p_2(0), \dots, p_r(0)$ being distinct nonzero constants. Say $\alpha_i = p_i(0)$. Wlog. let us assume that we want to construct an ABP for p_1 .

2. Applying homogenization.

We apply a particular kind of homogenization on the x_i variables, using a new variable z . This will give us an ABP computing $f' = z^c \prod_{i \in [r]} (q_i - y)$, for some integer $c > 0$, where $q_i = z^d p_i(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$ and $d = \max_i \deg(p_i)$. For $\alpha \in \mathbb{F}$, we consider *shifting* by αz^d , by defining $f'_\alpha = f'_{|y:=y+\alpha z^d} = \prod_{i \in [r]} (q_i - \alpha z^d - y)$. We want to exploit the fact that for the shifted polynomials $f'_{\alpha_2}, f'_{\alpha_3}, \dots, f'_{\alpha_r}$, since all α_i s are distinct, the factor $(q_1 - \alpha_i z^d - y)$ still contains the monomial z^d of highest z -degree, for every $2 \leq i \leq r$, whereas for every other factor this term is dropped in one of $f'_{\alpha_2}, f'_{\alpha_3}, \dots, f'_{\alpha_r}$.

3. Use the completeness of the determinant to obtain an eigenvalue problem.

For any $\alpha \in \mathbb{F}$, using the completeness of the determinant, we will obtain that $f'_\alpha = \det(P_\alpha - yQ)$, for some $\{0, -1\}$ -valued matrix Q and a *nonsingular* matrix P_α , whose entries are products of x_i variables and powers of z . Let $M_\alpha = \text{Adj}(P_\alpha)Q$, where $\text{Adj}(P_\alpha)$ denotes the *adjugate* of P_α , i.e. $P_\alpha^{-1} = \text{Adj}(P_\alpha) / \det(P_\alpha)$. It is not too difficult to see that for each i , $\lambda_i := \frac{\det(P_\alpha)}{q_i - \alpha z^d}$ is an eigenvalue of M_α , over the field of rational function $\mathbb{F}(X, z)$.

4. Selecting a starting point u , and applying power iteration.

Say for $i \in [r]$, v_i is an eigenvectors corresponding to λ_i , whose entries are *polynomials*. We take the matrix $V = [v_1, v_2, \dots, v_r, e_{i_1}, \dots, e_{i_{s-r}}]$, where we extend with some standard basis vectors e_{i_j} to make V nonsingular. For an arbitrary point $u \in \text{Range}(M_0)$, writing $u = a_1 v_1 + \dots + a_r v_r$, means we have to apply $V^{-1} = \text{Adj}(V) / \det(V)$ to u in order to express u in the different basis. Generally this is problematic, since then we obtain a_i s that are rational functions, rather than polynomials. To stress, *at any point of our computation, we want to make sure that we are computing with polynomials, so that at the end we can obtain (a multiple of) v_1 based on considering z -degrees of terms.* We will spend an important

part of the proof showing that $\det(V)^2$ can actually be computed without direct knowledge of the v_i s. Hence we can scale up the above a_i s by a factor of $\det(V)^2$, to ensure they are elements of $\mathbb{F}[X, z]$, and bootstrap the computation. In order to show this, we have to normalize our ABPs to some deliberately chosen standard form. Then we can provide a closed form for each v_i in terms of the normalized ABP and the (unknown) value q_i . It turns out that for eigenvectors v_1, v_2, \dots, v_r obtained this way, $\det(V)^2 = z^{c'} \prod_{1 \leq i < j \leq n} (q_i - q_j)^2$, for some integer $c' > 0$. Since $\prod_{1 \leq i < j \leq n} (x_i - x_j)^2$ is a symmetric polynomial, we know it is expressible in terms of the elementary symmetric polynomials $S_r^j(x_1, \dots, x_r) = \sum_{I \subset [r]} \prod_{i \in I} x_i$, due to the fundamental theorem on symmetric polynomials. Eventhough we cannot directly compute the q_i s, $S_r^j(q_1, \dots, q_r)$ equals the coefficient of $(-y)^{r-j}$ in f^j (ignoring the factor $z^{c'}$). We will see that therefore it can be obtained by some standard circuit manipulations. This way we obtain a relatively small ABP for $\det(V)^2$.

The next step is to apply power iteration. We construct a (multi-output) ABP computing $M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e u$, where e is some appropriately selected large integer. The next crucial part of the proof is to show that when changing α , only the eigenvalues of M_α change, but that the v_i s remain to be a valid set of eigenvectors. This will allow us to finally arrive at the following expression: $u' := M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e u = \sum_{i \in [r]} a_i \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i$. We will have that u' is a vector of *polynomials*, since we ensure that every $a_i \in \mathbb{F}[X, z]$, and that every v_j and M_α only contains polynomial entries. We will show that from the vector u' , we can separate out a multiple of the eigenvector v_1 by discarding terms that have z -degree larger than some threshold. To keep e reasonable we will need to provide good bounds on the degrees of the a_i s and entries of the eigenvectors v_j . Next, we can compute q_1 by applying M_0 once more and doing a division. For the latter, we use a recent result by Kaltofen and Koiran [7] to perform the (exact) division of two ABPs. Finally, the ABP for p_1 is obtained by setting $z = 1$ in the ABP for q_1 .

2 Preliminaries

Let \mathbb{F} be a field of characteristic zero. Let $X = \{x_1, x_2, \dots, x_n\}$ be a set of indeterminates. Let \mathbb{G} denote the field of rational functions $\mathbb{F}(X, z)$. For a poly-

nomial $f \in \mathbb{F}[X, y]$ and $p \in \mathbb{F}[X]$, $f_{|y=p}$ denotes the polynomial obtained by substituting p for y in f . In case $f_{|y=p} \equiv 0$, we say that p is a *root* of f for y . Recall the following lemma by Gauss:

Lemma 1. (Gauss). *Let $f \in \mathbb{F}[X, y]$ be a nonzero polynomial, and let $p \in \mathbb{F}[X]$ be a root of f for y . Then $p - y$ is an irreducible factor of f in the ring $\mathbb{F}[X, y]$.*

In the above situation, the *multiplicity* of the root p is defined to be the largest number m such that $(p - y)^m$ divides f . The Vandermonde determinant is the polynomial $(x_1, x_2, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_i - x_j)$. For two polynomials f and g , if there exists a polynomial h such that $f = gh$, we say g *divides* f , and that the division f/g is *exact*. Total degree of a nonzero polynomial f is denoted by $\deg(f)$. The maximum degree of a variable x_i in a monomial of a nonzero polynomial f is denoted by $\deg_{x_i}(f)$. For a nonzero vector v of polynomials, we define $\deg(v) = \max_{j, v_j \neq 0} \deg(v_j)$ and $\deg_{x_i}(v) = \max_{j, v_j \neq 0} \deg_{x_i}(v_j)$. For a polynomial $f \in \mathbb{F}[X]$, we denote by $[f]_{=i}$, or simply $f_{=i}$, the homogeneous component of degree i . Similarly, we use the notations $f_{\leq i}$, $f_{\geq i}$, $[f]_{\leq i}$ and $[f]_{\geq i}$. We will also use this notation for vectors of polynomials. For example, $(f, g, h)_{\leq i} = (f_{\leq i}, g_{\leq i}, h_{\leq i})$. At a few occasions we will also use the notation $f_{\leq_i i}$, for $f \in \mathbb{F}[X, z]$. This is defined analogously, but now using the individual degree measure \deg_z instead of total degree measure \deg . For a nonzero polynomial f , $\mindeg(f)$ is the minimum i such that $f_{=i}$ is nonzero. Similarly, we define $\mindeg_{x_j}(f)$ to be the minimum i such that x_j^i appears in a monomial of f . We extend this to any nonzero vector v of polynomials, by letting $\mindeg(v) = \min_{j, v_j \neq 0} \mindeg(v_j)$ and $\mindeg_{x_i}(v) = \min_{j, v_j \neq 0} \mindeg_{x_i}(v_j)$. Given an integer $d \geq 0$ and a variable z , for a polynomial f of degree $\leq d$, the *homogenization of f to degree d using the variable z* , is the polynomial $z^d f(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$.

For a matrix M , we denote by $M[[i, j]]$ the matrix obtained by removing row i and column j . Let us denote by $M[i, j]$ the matrix obtained from M by setting all entries in row i and column j to 0, except for entry $M_{i,j}$, which is set to 1. Using Laplace expansion along row i of $M[i, j]$ one immediately concludes that the following holds: for any matrix M , $\det(M[i, j]) = (-1)^{i+j} \det(M[[i, j]])$. For a $s \times s$ matrix M , $Adj(M)$ denotes the $s \times s$ *adjugate* matrix of M , defined by $Adj(M)_{ij} = \det(M[[j, i]])$. For any matrix M , $MAdj(M) = Adj(M)M = \det(M)I$, where I denotes the identity matrix.

An *arithmetic circuit* Ψ over variables X and field \mathbb{F} is given by a directed acyclic graph whose nodes of in-degree larger than zero are labeled by $\{+, \times\}$, and with other nodes labeled by elements of $X \cup \mathbb{F}$. At each node g of Ψ we have associated a polynomial in $\mathbb{F}[X]$ computed by g , which is defined in the standard manner. The output of Ψ is the polynomial computed by some designed output gate. For the size of Ψ we count the number of edges. The size of a polynomial f , denoted by $s(f)$, is the size of the smallest arithmetic circuit computing f . We let VP stand for the class of polynomial families (f_n) for which there exists a polynomial $p(n)$ such that $\deg(f_n) \leq p(n)$ and $s(f_n) \leq p(n)$.

An algebraic branching program (ABP) over X and \mathbb{F} is a 4-tuple $\Phi = (G, w, \sigma, \tau)$, where $G = (V, E)$ is a weighted directed acyclic graph for which the vertex set V can be partitioned into *levels* L_0, L_1, \dots, L_ℓ , where $L_0 = \sigma$ and $L_\ell = \tau$. Vertices σ and τ are called the source and sink of Φ , respectively. Edges may only go between consecutive levels L_i and L_{i+1} . The subgraph induced by L_i and L_{i+1} is called a *layer* of Φ . The weight function $w : E \rightarrow X \cup \mathbb{F}$ assigns variables or field constants to the edges of G . For a path p in G , we extend the weight function by $w(p) = \prod_{e \in p} w(e)$. Let $P_{i,j}$ denote the collection of all directed paths p from i to j in G . The program Φ computes the polynomial $\hat{\Phi} := \sum_{p \in P_{\sigma, \tau}} w(p)$. The size of Φ is defined to be $|V|$. For nodes v and w in Φ , $\Phi_{v,w}$ denotes the subprogram of Φ with source v and sink w .

We extend the definition to what we call *generalized* ABPs as follows. Let z be a new variable. Let $\mathcal{W} = \{z^d \cdot \ell : d \geq 0, \ell \in X \cup \mathbb{F}\}$. For generalized ABPs, we allow any weight $w(e)$ to be an element in \mathcal{W} . We will also consider ‘‘multi-output’’ ABPs. In this case the last layer of the A consists of several sink nodes $\tau_1, \tau_2, \dots, \tau_m$. The output of the ABP is given by the tuple of polynomials (f_1, f_2, \dots, f_m) computed by the subprograms $A_{\sigma, \tau_1}, A_{\sigma, \tau_2}, \dots, A_{\sigma, \tau_m}$.

ABPs are convenient when dealing with substitution. It is easily seen that if g can be computed by an ABP A_g of size s_g and f is computed by an ABP A_f of size s_f , then $f|_{x_i=g}$ can be computed by an ABP of size $O(e_f s_g)$, where $e_f = O(s_f^2)$ is the number of edges in A_f . For the analysis, we define absolute constants $\gamma_1 = 3, \gamma_2 = 5, \gamma_3 = 12$. We use a result by Mahajan and Vinay and a result by Kaltofen and Koiran.

Theorem 3 (See Theorem 2 in [8]). *The determinant of an $n \times n$ matrix can be computed by an ABP of size $O(n^{\gamma_1})$ with $O(n^{\gamma_2})$ many edges.*

Lemma 2 (See⁴[7]). *Suppose $|\mathbb{F}|$ is infinite. Let $f, g \in \mathbb{F}[X]$ be given that both are computable by ABPs of size at most s . Assuming the division f/g is exact, then f/g can be computed by an ABP of size $O(s^{\gamma_3})$.*

The following two lemma’s are proved by the well-known trick of ‘splitting’ nodes in order to keep track of degree components.

Lemma 3. *Let $d \geq 0$ be an integer and z a new variable. Let Φ be an ABP of size s computing the polynomial $f \in \mathbb{F}[X]$ of degree at most d . Then there exist an ABP Ψ of size $O(ds)$ computing the homogenization of f to degree d using the variable z .*

Lemma 4. *Let $d \geq 0$ be an integer and z a new variable. Let Φ be a generalized ABP of size s with e many edges computing the polynomial $f \in \mathbb{F}[X, z]$. Then there exist an ABP Ψ of size $O(ds)$ and $O(de)$ many edges computing $[f]_{\leq d}$. A similar statement holds for computing $[f]_{=z^d}$.*

Finally, we use a result by Kaltofen and Singer for computing formal partial derivatives. We use the notation $\frac{\partial^k f}{\partial^k y}$ to denote the formal partial derivative of f of order k w.r.t. the variable y .

Theorem 4 (Theorem 3.1 in [10]). *For any integer $k \geq 0$, if $f \in \mathbb{F}[X, y]$ can be computed by an arithmetic circuit of size s , then $\frac{\partial^k f}{\partial^k y}$ can be computed by an arithmetic circuit of size $O(\mathcal{M}(k) \cdot s)$.*

3 Reduction to root extraction for ‘nice’ polynomials

Two polynomials $p, q \in \mathbb{F}[X]$ are said to be in *general position*, if $p(0)$ and $q(0)$ are both nonzero, and $p(0) \neq q(0)$. A set of polynomials $\{p_1, p_2, \dots, p_r\}$ is said to be in general position, if for every $i, j \in [r]$ with $i \neq j$, p_i and p_j are in general position. A polynomial $f \in \mathbb{F}[X, y]$ is called *nice*, if f factors as $f = (p_1 - y)(p_2 - y) \dots (p_r - y)$, where $\{p_1, p_2, \dots, p_r\} \subset \mathbb{F}[X]$ is a set of nonconstant polynomials that is in general position.

Suppose for the above situation we have a method of constructing an ABP for any p_i that is of size at most $\beta(r, d, s)$ for some function β , where s denotes the ABP size of f and $d = \max_i \deg(p_i)$. We reduce the

⁴This follows from From Lemma 1 and Lemma 2 in [7]. We get an extra quadratic blow-up of s , since the DAG we use for ABPs must be leveled.

more general case of Main Theorem 1 to root extraction for nice polynomials with the following lemma.

Lemma 5. *Suppose $|\mathbb{F}|$ is infinite. Suppose a nonzero polynomial $f \in \mathbb{F}[X, y]$ factors as $f = p_0(p_1 - y)(p_2 - y) \dots (p_r - y)$, where $\{p_0, p_1, p_2, \dots, p_r\} \subset \mathbb{F}[X]$ and p_1, p_2, \dots, p_r are pairwise distinct. If f can be computed by an ABP of size s , then every p_i has an ABP of size $O(\beta(r, d, O(r^{\gamma_3} s^{\gamma_3})))$, where $d = \max_{i \in [r]} \deg(p_i)$ and γ_3 is the absolute constant from Section 2.*

Proof. By Lemma 4, we have an ABP for the coefficient of y^r in f of size $O(rs)$. This program computes $(-1)^r p_0$. If there are p_i s that are constant, any of these can be computed by ABPs with size at most 2. Wlog. assume p_1, p_2, \dots, p_j are constant. Since $j \leq r \leq s$, it is easily seen we have an ABP of size $O(rs)$ that computes $p_0(p_1 - y)(p_2 - y) \dots (p_j - y)$. Now use Lemma 2 to obtain an ABP of size $s = O(r^{\gamma_3} s^{\gamma_3})$ computing $\tilde{f} = (p_{j+1} - y)(p_{j+2} - y) \dots (p_r - y)$. Hence at the cost of blowing up the size to $O(r^{\gamma_3} s^{\gamma_3})$, we can assume that f is of the form $f = (p_1 - y)(p_2 - y) \dots (p_r - y)$, where p_1, p_2, \dots, p_r are nonconstant and pairwise distinct.

Since $|\mathbb{F}|$ is infinite, there exists $a \in \mathbb{F}^n$ such that for every i , $p_i(a) \neq 0$, and for every $i \neq j$, $p_i(a) \neq p_j(a)$. Namely, we can simply take a nonzero of the polynomial $\prod_{i \in [r]} p_i \prod_{i \neq j \in [r]} (p_i - p_j)$. Consider $f' := f(x_1 + a_1, x_2 + a_2, \dots, x_n + a_n, y)$. We have that $f' = \prod_{i=1}^r (p_i(x_1 + a_1, x_2 + a_2, \dots, x_n + a_n) - y) = \prod_{i=1}^r (p_i(x_1 + a_1, x_2 + a_2, \dots, x_n + a_n)_{\geq 1} + p_i(a_1, a_2, \dots, a_n) - y)$. Hence f' is nice. An ABP of size $O(r^{\gamma_3} s^{\gamma_3})$ for f' is easily obtained from the ABP for f . We can then do the root extraction for the nice polynomial f' . This gives us an ABP for any desired $p_i(x_1 + a_1, x_2 + a_2, \dots, x_n + a_n)$ of size at most $\beta(r, d, O(r^{\gamma_3} s^{\gamma_3}))$. Next we easily perform a modification of this program to realize the substitution $x_i := x_i - a_i$, for all $i \in [r]$, while blowing up the size by a constant factor at most. Hence, we obtain an ABP for $p_i(x_1, x_2, \dots, x_n)$ of size $O(\beta(r, d, O(r^{\gamma_3} s^{\gamma_3})))$. \square

4 Standard form ABPs, Valiant matrices and homogenizations

Definition 1. *Let $f \in \mathbb{F}[X, y]$ be a polynomial whose degree in y equals r , and write $f = \sum_{i=0}^r C_r(x)y^i$. We say an ABP Φ with source σ and sink τ computing f is in standard form, if it has the following structure:*

- There is a set of distinct nodes $\{b_0, b_1, \dots, b_r\}$, such that for each $i \in \{0, 1, \dots, r\}$, there is an edge from the source σ to b_i with label 1. These are the only edges adjacent to the source.
- There are distinct nodes c_0, c_1, \dots, c_r . The subprograms in the set $\{\Phi_{b_i, c_i} : i \in [r]\}$ are disjoint as graphs. For every $i \in \{0, 1, \dots, r\}$, the subprogram⁵ Φ_{σ, c_i} computes $C_i(x)$.
- There is a path $c_r = a_0, a_1, \dots, a_{r-1}, a_r = \tau$, where each edge (a_i, a_{i+1}) is labeled with the variable y . These are the only occurrences of y variables in Φ .
- All remaining edges are labeled with the constant one. These simply realize that for every $0 \leq i < r$, there is one single path of weight 1 from c_i to a_{r-i} .
- the length of every path from σ to τ is even.

More generally, if in the above edges not labeled with y carry labels $\in \mathcal{W}$, then we say that Φ is in generalized standard form.

The following lemma follows from Lemma 4:

Lemma 6. *Let $f \in \mathbb{F}[X, y]$ be computed by an ABP Φ of size s , and let $r = \deg_y(f)$. Then f can be computed by an ABP Ψ in standard form of size $O(sr^2)$. This means in particular that the variable y appears exactly r times on an edge in Ψ .*

Given an ABP Φ of size s computing f , we can construct a matrix $M(\Phi)$ of order s , whose entries are variables and field elements, such that $\det(M) = f$, as done in [11]. Namely, thinking of Φ as a graph, one adds a loop back from τ to σ with label 1, and one puts a self loop on all nodes other than σ and τ with label 1. Let $M(\Phi)$ be the⁶ adjacency matrix of the weighted graph obtained this way, which we call the *Valiant matrix associated to Φ* . Assuming wlog. that the length of every path from σ to τ in Φ is even, then $\det(M(\phi)) = f$. Rows and columns of $M(\Phi)$ correspond to nodes in Φ . In our notation, we will use variable names of nodes of Φ to index the matrix $M(\Phi)$, and also do this for s -vectors operated on. For example, for the standard form ABP Φ for Definition 1, the entry $M(\Phi)_{a_0 a_1}$ equals y . More generally, if we start with a generalized ABP Φ , we define in a completely analogous fashion the associated *general-*

⁵For a standard form ABP Φ_{b_i, c_i} and Φ_{σ, c_i} compute the same polynomial. For generalized standard form ABPs the computed polynomials differ by a factor of $w(\sigma, b_i)$.

⁶Wlog. we can assume nodes in Φ always carry a unique number $\in [s]$, which we then use to index columns/rows. This way, we can truly speak of the matrix $M(\Phi)$.

ized Valiant matrix $M(\Phi)$. In this case, $M(\Phi)$ contains elements from \mathcal{W} .

Definition 2. Let z be a variable. Given the Valiant matrix $M(\Phi)$ associated to an ABP Φ , we define its d -homogenization to be the matrix obtained from $M(\Phi)$ by

1. Replacing every variable entry x_i by $x_i z^{d-1}$.
2. Replacing every constant entry $c \in \mathbb{F}$ by cz^d .
3. Leaving y variables unchanged.

We denote this matrix by $\overline{M(\Phi)}$, provided it is clear from the context what d is. Then we can write $\overline{M(\Phi)} = z^d M(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}, \frac{y}{z^d})$. For an ABP Φ , its d -homogenization $\overline{\Phi}$ is the generalized ABP obtained by performing the above replacement operations (1,2,3) for every edge label.

The following proposition is left as an easy exercise:

Proposition 1. Let Φ be an ABP of size s with source σ and sink τ . For the case of d -homogenization, we have that

1. $M(\overline{\Phi})$ and $\overline{M(\Phi)}$ only differ for nonzero entries on the diagonal and the ‘loopback’ entry on row τ , column σ . For these entries, $M(\overline{\Phi})$ contains the field element 1, whereas $\overline{M(\Phi)}$ contains z^d .
2. $\det(\overline{M(\Phi)}) = z^{(s-\ell)d} \det(M(\overline{\Phi}))$, where ℓ equals the number of layers of Φ .

The following is now easy to prove:

Proposition 2. Suppose Φ is an ABP of size s computes $f \in \mathbb{F}[X, y]$, where f factors as $f = \prod_{i \in [r]} (p_i - y)$. Let $d = \max_i \deg(p_i)$. Then for d -homogenization $\overline{M(\Phi)}$ we have that $\det(\overline{M(\Phi)}) = z^{d(s-r)} \prod_{i \in [r]} (q_i - y)$, where $\forall i \in [r]$, $q_i = z^d p_i(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$.

Proof.

$$\begin{aligned}
 & \det(\overline{M(\Phi)}) \\
 &= \det(z^d M(\Phi)(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}, \frac{y}{z^d})) \\
 &= z^{ds} \det(M(\Phi)(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}, \frac{y}{z^d})) \\
 &= z^{ds} f(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}, \frac{y}{z^d}) \\
 &= z^{ds} \prod_{i \in [r]} (p_i(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z}) - \frac{y}{z^d}) \\
 &= z^{d(s-r)} \prod_{i \in [r]} (q_i - y).
 \end{aligned}$$

□

4.1 A closed form for eigenvectors related to the Valiant matrix

The following proposition is easily proved using Laplace expansion:

Proposition 3. Let M be a singular matrix of order m . For any fixed i , if we define the m -vector v by taking $v_j = \det(M[i, j])$, then $Mv = 0$.

Next we derive the main lemma of this subsection.

Lemma 7. Let Φ be a generalized ABP of size s in standard form computing the polynomial $f = \sum_{i=0}^r C_i(X, z)y^i \in \mathbb{F}[X, z, y]$ of degree r in y . Let d be a bound on the z -degree of edge labels in Φ . Let nodes $a_0, b_0, c_0, a_1, b_1, c_1, \dots, a_r, b_r, c_r$ be given as in Definition 1, which implies the subprogram Φ_{σ, c_i} computes $C_i(X, z)$. Let $M(\Phi)$ be the associated generalized standard form Valiant matrix. Suppose that $q \in \mathbb{F}[X, z]$ is such that $f_{|y=q} \equiv 0$. Let N be the matrix obtained by setting $y = q$ in $M(\Phi)$, i.e. $N = M(\Phi)_{|y=q}$. Suppose we define the s -vector v by $(v)_j = \det(N[c_r, j])$, for all $j \in [s]$. Then the following hold:

1. $Nv = 0$.
2. $\deg_z(\det(N[c_i, j])) \leq s \max(\deg_z(q), d)$.
3. $\forall j \in [r]$, $(v)_{a_j} = q^{r-j} \cdot (-1)^{j+1} C_r(X, z)$.

Proof. We recall the notion of a cycle cover for later use. A cycle cover C in a directed graph $G = (V, E)$ with n vertices is a set of disjoint simple cycles C_1, C_2, \dots, C_i such that every vertex in G is contained in some cycle C_i . For weighted G , the weight of a cycle C is taken to be the product of weights of edges in C . For a simple cycle C we define its sign $\text{sgn}(C)$ to be -1 if C is of even length, and 1 otherwise. For the cycle cover C , define $\text{sgn}(C) = \prod_i \text{sgn}(C_i)$.

Observe that $\det(N) = \det(M(\Phi)_{|y=q}) = f_{|y=q} \equiv 0$. Hence the first property follows from Proposition 3. The second property is clear. To verify the last property, let $j \in [r]$ be arbitrary. Consider the matrix $N = M(\Phi)_{|y=q}$. Let G be the weighted graph corresponding to $M(\Phi)$. We can think of the matrix $N[c_r, a_j]$ as the adjacency matrix of a graph H formed by doing the following to G :

- replacing all y -labels in G by q .
- Removing all edges out of c_r , including the self loop.
- Removing all edges into a_j , including the self loop.
- Adding the edge From c_r to a_j with label one.

Then $\det(N[c_r, a_j]) = \sum_C \text{sgn}(C)w(C)$, where the sum is over all cycle covers in H . Observe that since c_r and a_j do not have self-loops, any cycle cover C in H must include the edge (c_r, a_j) . So the cycle covers are of the following structure: 1) From the source σ there is a path to c_r , 2) The edge (c_r, a_j) is taken, 3) $r - j$ edges with label q are taken, 4) Finally, the loop back from $a_r = \tau$ to the source σ is taken, and self-loops with label 1 are taken for all vertices not included in above cycle.

All of the above described cycles starting at the source σ are of the same length. In case $j = 1$ the length equals the same ‘big cycle length’ as in $M(\Phi)$, which is odd. For general j , by considering how many edges we skip with the edge (c_r, a_j) one can conclude that $\text{sgn}(C) = (-1)^{j+1}$. Hence $\det(N[c_r, a_j]) = \sum_C \text{sgn}(C)w(C) = (-1)^{j+1} \sum_C w(C)$. The expression $\sum_C w(C)$ equals the sum of weights of all paths from σ to τ that go over (c_r, a_j) . Since these paths all go over c_r , this sum factors as $C_r(X, z)$ (weight of all paths from σ to c_r) times q^{r-j} (weights of path “ (c_r, a_j) , followed by going from a_j to τ ”). \square

5 Proof of the main theorem

Lemma 8. *Let $f \in \mathbb{F}[X, y]$ be a nice polynomial of degree $r > 1$ computed by a standard form ABP Φ of size s . Suppose f factors as $f = \prod_{i \in [r]} (p_i - y)$, where $\{p_1, p_2, \dots, p_r\} \subset \mathbb{F}[X]$ is a set of nonconstant polynomials in general position. Then any p_i can be computed by an ABP of size $O((r^3 d^7 s^{12+\gamma_1} + r^{6+\gamma_2} d^5 s^7)^{\gamma_3})$, where $d = \max_{i \in [r]} \deg(p_i)$ and γ_1, γ_2 and γ_3 are the absolute constants introduced in Section 2.*

Proof. Wlog. we will show the method for obtaining p_1 .

5.1 Towards computing eigenvectors

Consider the associated Valiant matrix $M(\Phi)$. Let $\overline{M(\Phi)}$ be the d -homogenization of $M(\Phi)$. Note that $f|_{y=0} \neq 0$. Write $M(\Phi) = A - yB$, by taking B to have -1 precisely in the places where $M(\Phi)$ has y 's and zeros everywhere else. Then $f = \det(A - yB)$. We have the following two properties:

1. In each row/column there is at most one -1, and the number of -1s in B equals r .
2. A is invertible over $\mathbb{F}(X)$.

Let $q_i = z^d p_i(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$. Each polynomial q_i is homogeneous of degree d . Restricting our attention to degrees in z only, we see that the original constant term $p_i(0)$ of p_i is mapped to the term $p_i(0)z^d$ in q_i with largest z -degree. Let $\alpha_i = \text{coef}(q_i, z^d)$. We have that $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ is a set of r distinct nonzero values from \mathbb{F} . Note that $s \geq r$. Define $f_0 = z^{d(s-r)}(q_1 - y)(q_2 - y) \dots (q_r - y)$. More generally, for any $\alpha \in \mathbb{F}$, define $f_\alpha = (f_0)|_{y:=y+\alpha z^d}$. Then $f_\alpha = z^{d(s-r)} \cdot \prod_{i \in [r]} (q_i - \alpha z^d - y)$. Let $R = z^d A(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$, and $Q = B$.

Lemma 9. *The following statements hold:*

1. $R - yQ = \overline{M(\Phi)}$.
2. $\forall \alpha \in \mathbb{F}, f_\alpha = \det(R - \alpha z^d Q - yQ)$.
3. R is nonsingular.

Proof. The first property is clear. For the second property, note that Proposition 2 gives that $f_0 = \det(\overline{M(\Phi)})$. So $f_0 = \det(R - yQ)$, by the first property. Hence $f_\alpha = \det(R - yQ)|_{y:=y+\alpha z^d} = \det(R - \alpha z^d Q - yQ)$. The third property follows from the second property. Namely, $(f_0)|_{y=0} = \det(R)$. It must be that $(f_0)|_{y=0} \neq 0$, since otherwise $q_i = 0$, for some i . However, this means that $p_i = 0$, which is a contradiction. \square

Let ℓ be the number of layers of Φ . Note that $s > \ell > r$. Define for any $\alpha \in \mathbb{F}, \overline{f}_\alpha = f_\alpha / z^{d(s-\ell)}$. Then $\overline{f}_\alpha = z^{d(\ell-r)} \cdot \prod_{i \in [r]} (q_i - \alpha z^d - y)$. Let $\overline{\Phi}$ be the d -homogenization of Φ . Note that $\overline{\Phi}$ is in generalized standard form. Observe, that Q does not have nonzero entries on its diagonal or on the ‘loopback’ entry on row τ , column σ , since in the standard form nodes labeled with the variable y do not appear on self-loops or the loopback edge from τ to σ . By Proposition 1, we can write $M(\overline{\Phi}) = P - yQ$, where P is obtained from R by setting all nonzero diagonal entries and the ‘loopback’ entry (τ, σ) to the field element 1.

Corollary 1. *We have the following two properties:*

1. For all $\alpha \in \mathbb{F}, \det(P - \alpha z^d Q - yQ) = \overline{f}_\alpha$.
2. For every $\alpha \in \mathbb{F}, P - \alpha z^d Q$ is nonsingular.

Proof. From Proposition 1, it follows that $\overline{f}_0 = \det(M(\overline{\Phi})) = \det(P - yQ)$. This readily gives the first stated property. From this we conclude that $\det(P - \alpha z^d Q) = (\overline{f}_\alpha)|_{y=0}$. If $(\overline{f}_\alpha)|_{y=0}$ is zero, then there exists i such that $q_i - \alpha z^d$ equals zero. This implies p_i was a constant polynomial, which is a contradiction. \square

Note that $\bar{\Phi}$ computes \bar{f}_0 .

Lemma 10. *Let nodes $\{a_i, b_i, c_i : i \in [r]\}$ in $\bar{\Phi}$ be given as in Definition 1. Let $C_i(X, z)$ be the polynomial computed by the subprogram $\bar{\Phi}_{\sigma, c_i}$. Clearly, we have that $\bar{f}_0 = \sum_{i=0}^r C_i(X, z)y^i$. For all $i \in [r]$, we define the column vector v_i by letting for every $j \in [s]$, $(v_i)_j = \det(N_{q_i}[c_r, j])$, where $N_{q_i} = N_{q_i} = P - q_i Q$. Then the following hold:*

1. For every $i \in [r]$, $N_{q_i} v_i = 0$.
2. For every $i \in [r]$, $\deg_z(v_i) \leq sd$.
3. There exist standard basis vectors $e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}$ such that for $V := [v_1, v_2, \dots, v_r, e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}]$, $\det(V) = (\pm 1) \cdot z^{d(\ell-r)r} \cdot \text{Vandet}(q_1, q_2, \dots, q_r)$.
4. $\text{Vandet}(q_1, q_2, \dots, q_r)^2$ can be computed by an ABP of size $O(r^{5+\gamma_2} ds)$, where γ_2 is the absolute constant introduced in Section 2.

Proof. The first and second item immediately follow from Lemma 7, Items 1 and 2. Note that $C_r(X, z) = \text{coef}(y^r, \bar{f}_0) = (-1)^r z^{d(\ell-r)}$. By Lemma 7, Item 3, up to reordering of rows and multiplying rows with the field element -1 , the matrix $V' = [v_1, v_2, \dots, v_r]$ consisting of the r column vectors v_1, v_2, \dots, v_r contains the $r \times r$ matrix $z^{d(\ell-r)} \cdot U$ as a submatrix on rows in the set $J = \{a_j : j \in [r]\}$, where U is the Vandermonde matrix $(q_i^j)_{0 \leq j \leq r-1, 1 \leq i \leq r}$. Choosing $e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}$ to be an independent set of vectors that is zero on rows indexed by J gives the third property. The proof of the fourth property we treat in the next subsection. \square

5.2 A small ABP for computing $\text{Vandet}(q_1, q_2, \dots, q_r)^2$

This subsection is dedicated to proving Item 4 of Lemma 10. Define the polynomial

$$T_i(x_1, x_2, \dots, x_r) = x_1^i + x_2^i + \dots + x_r^i.$$

We use the fact⁷ that

$$\det \begin{pmatrix} T_0 & T_1 & \dots & T_{r-1} \\ T_1 & T_2 & \dots & T_r \\ \vdots & \vdots & \ddots & \vdots \\ T_{r-1} & T_r & \dots & T_{2r-2} \end{pmatrix} \quad (1)$$

$$= \text{Vandet}(x_1, x_2, \dots, x_r)^2. \quad (2)$$

The strategy is to express each T_i as a ‘small’ formula

⁷This follows by multiplying the Vandermonde matrix with nodes x_1, x_2, \dots, x_r by its transpose.

of $S_r^j(x_1, x_2, \dots, x_r)$, where S_r^j is the elementary symmetric polynomial in r variables of degree j , i.e.

$$S_r^j(x_1, x_2, \dots, x_r) = \sum_{I \subset [r], |I|=j} \prod_{i \in I} x_i.$$

It is well-known that the T_i s and S_r^j s are related through the Newton Identities.

At first sight it may look like we have run into a circular argument. How do we plug in the q_i s? This bootstrapping problem is resolved by observing that, if we succeed in the above⁸, regardless of not having small ABPs for the q_i s, we readily have small ABPs for any $S_r^j(q_1, q_2, \dots, q_r)$. Namely, consider the following remark and subsequent derivation:

Remark 1. *For every j , $S_r^j(p_1, p_2, \dots, p_r)$ equals the coefficient of y^{r-j} of f modulo a factor of ± 1 . Hence an ABP Φ_j computing $S_r^j(p_1, p_2, \dots, p_r)$ of size at most s is easily obtained from the standard form ABP Φ .*

Say that Φ_j computes the polynomial $D_j(X, z)$. We conclude that we have an ABP Ψ_j computing $S_r^j(q_1, q_2, \dots, q_r)$ of size $O(rds)$ as follows:

$$\begin{aligned} S_r^j(q_1, q_2, \dots, q_r) &= \\ S_r^j(z^d p_1(\frac{x_1}{z}, \dots, \frac{x_n}{z}), \dots, z^d p_r(\frac{x_1}{z}, \dots, \frac{x_n}{z})) &= \\ z^{dj} S_r^j(p_1(\frac{x_1}{z}, \dots, \frac{x_n}{z}), \dots, p_r(\frac{x_1}{z}, \dots, \frac{x_n}{z})) &= \\ z^{dj} (\pm 1) \cdot D_j(\frac{x_1}{z}, \dots, \frac{x_n}{z}). \end{aligned}$$

Note that the degree of D_j is at most dj . So $z^{dj} D_j(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$ is just the homogenization of D_j to degree dj . Applying Lemma 3 to Φ_j yields the required ABP Ψ_j .

5.2.1 A formal power series identity related to the Newton identities

Let w be a new variable. We have the following lemma:

Lemma 11. *Provided the characteristic of \mathbb{F} is zero, we have the following identity in the ring of formal power series $\mathbb{F}[[X]]$: $\sum_{\ell \geq 1} \frac{1}{\ell} (\sum_{j=1}^r (-1)^j S_r^j(x_1, x_2, \dots, x_r) w^j)^\ell = \sum_{n \geq 1} \frac{-w^n}{n} T_n(x_1, x_2, \dots, x_r)$.*

Proof. We recall the definitions of the formal power series (FPS) $\text{exp}(w)$ and $\log(1-w)$. These are given by

⁸In [12] the converse is achieved to get small depths formulas for S_r^j .

$\exp(w) = \sum_{n \geq 0} \frac{w^n}{n!}$. and $\log(1 - w) = -\sum_{n \geq 1} \frac{w^n}{n}$. We will use that $\exp(\log(1 - wx_j)) = 1 - wx_j$. Hence

$$\begin{aligned} & \prod_{j \in [r]} (1 - wx_j) \\ &= \prod_{j \in [r]} \exp(\log(1 - wx_j)) \\ &= \exp\left(\sum_{j \in [r]} \log(1 - wx_j)\right) \\ &= \exp\left(-\sum_{j \in [r]} \sum_{n \geq 1} (wx_j)^n / n\right) \\ &= \exp\left(-\sum_{n \geq 1} T_n(x_1, x_2, \dots, x_r) w^n / n\right) \end{aligned}$$

Hence, by multiplying out the l.h.s. we get that

$$\begin{aligned} & \sum_{j=1}^r (-1)^j S_n^j(x_1, x_2, \dots, x_r) w^j = \\ & \exp\left(-\sum_{n \geq 1} T_n(x_1, x_2, \dots, x_r) w^n / n\right) - 1. \end{aligned}$$

Now we use that for $g(w) := -\sum_{n \geq 1} T_n(x_1, x_2, \dots, x_r) w^n / n$, it holds that $\log(1 + (\exp(g(w)) - 1)) = g(w)$. Thus applying $\log(1 + w)$ to both sides of the above equation yields that

$$\begin{aligned} & -\sum_{n \geq 1} T_n(x_1, x_2, \dots, x_r) w^n / n = \\ & \log\left(1 - \sum_{j=1}^r (-1)^j S_n^j(x_1, x_2, \dots, x_r) w^j\right) = \\ & \sum_{\ell \geq 1} \frac{1}{\ell} \left(\sum_{j=1}^r (-1)^j S_n^j(x_1, x_2, \dots, x_r) w^j\right)^\ell. \end{aligned}$$

□

In the following, we truncate the expression on the l.h.s. in the above lemma, discarding terms that cannot possibly contribute to the coefficient of w^i . Then we do some circuit manipulations to extract the coefficient of w^i , and this way we obtain an ABP computing T_i in terms of the S_r^j s.

Proposition 4. *Let u_1, u_2, \dots, u_r be a set of new variables. For any $i \in [r]$ the following statements are true. Let $E(u_1, u_2, \dots, u_i, w) = \sum_{1 \leq \ell \leq i} \frac{1}{\ell} \left(\sum_{j=1}^i (-1)^j u_j w^j\right)^\ell$. Then*

1. *There exists an ABP $\Gamma(u_1, u_2, \dots, u_i, w)$ with $O(i^3)$ many edges computing E .*
2. *There exists an ABP $\Gamma'(u_1, u_2, \dots, u_i)$ with $O(i^4)$ many edges computing the coefficient of w^i in E .*

3. *Say Γ' computes the polynomial E' . Then*

$$\begin{aligned} & E'(S_r^1(x), S_r^2(x), \dots, S_r^i(x)) \\ &= -T_i(x_1, x_2, \dots, x_r) / i. \end{aligned}$$

Proof. The first item is left as an easy exercise. Then second item then follows by applying Lemma 4. The last item follows from Lemma 11. □

5.2.2 Putting it together

By Proposition 4,

$$\begin{aligned} & E'(S_r^1(q_1, \dots, q_r), \dots, S_r^i(q_1, \dots, q_r)) \\ &= -T_i(q_1, \dots, q_r) / i. \end{aligned}$$

By Remark 1 and comments thereafter, we conclude that for any $i \in [r]$, we have an ABP computing $T_i(q_1, \dots, q_r)$ of size $O(r^5 ds)$. The $r \times r$ determinant can be computed by an ABP with $O(r^{\gamma_2})$ many edges by Theorem 3. Hence using Equation (2) we obtain an ABP for computing $\text{Vandet}(q_1, q_2, \dots, q_r)^2$ of size $O(r^{5+\gamma_2} ds)$. □

5.3 Selecting a good starting vector u

Let $M_0 = \text{Adj}(P)Q$, and let $s \times s$ matrix $V = [v_1, v_2, \dots, v_r, e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}]$ be given by Lemma 10. The set $\{M_0 v : v \in \mathbb{G}^s\}$ we denote by $\text{Range}(M_0)$. Consider the following proposition:

Proposition 5. *Working over the field \mathbb{G} , we have*

1. *For every $i \in [r]$, v_i is an eigenvector of M_0 corresponding to the eigenvalue $\frac{\det(P)}{q_i}$.*
2. *v_1, v_2, \dots, v_r form a basis of $\text{Range}(M_0)$.*

Proof. By Corollary 1, the polynomials q_1, q_2, \dots, q_r are precisely all the solutions for y of the equation $\det(P - yQ) \equiv 0$. For a polynomial q , we have that $\det(P - q \cdot Q) \equiv 0 \Leftrightarrow \exists v \in \mathbb{G}^s \neq \vec{0}$, such that $Pv = q \cdot Qv$. Lemma 10 shows that for every $i \in [r]$, $(P - q_i Q)v_i = 0$. Due to Item 2 of Corollary 1, $\text{Adj}(P)$ is nonsingular. Hence this is equivalent to $(\det(P)I - q_i M_0)v_i = 0$. Since $q_i \neq 0$, we can rewrite this as $(\frac{\det(P)}{q_i}I - M_0)v_i = 0$. Hence v_i is an eigenvector of M_0 corresponding to eigenvalue $\frac{\det(P)}{q_i}$. Lemma 10 gives that v_1, v_2, \dots, v_r are independent vectors. Note that $\text{rank}(Q) = r$. Since $\text{Adj}(P)$ is nonsingular, we have that $\text{rank}(M_0) = \text{rank}(Q) = r$. Hence it must be that v_1, v_2, \dots, v_r form a basis of $\text{Range}(M_0)$. □

The following lemma now follows easily:

Lemma 12. $\exists i \in [s]$ such that $\det(V)^2 M_0 e_i = a_1 v_1 + a_2 v_2 + \dots + a_r v_r$, where 1) $\forall i, a_i \in \mathbb{F}[X, z]$, 2) $a_1 \neq 0$, and 3) $\forall i, \deg_z(a_i) \leq d^3 s^5$.

Proof. By Item 2 of Proposition 5, v_1, v_2, \dots, v_r forms a basis of $\text{range}(M_0)$. Hence for every e_i , we can write $\det(V)^2 M_0 e_i = a_{1,i} v_1 + a_{2,i} v_2 + \dots + a_{r,i} v_r$, for certain $a_{1,i}, a_{2,i}, \dots, a_{r,i} \in \mathbb{G}$. Suppose that for every $i \in [s]$, $a_{1,i} = 0$. This means that $\text{range}(M_0) \subseteq \text{span}(v_2, \dots, v_r)$, i.e. $\text{rank}(M_0) \leq r-1$. This is a contradiction, as we observed before that $\text{rank}(M_0) = r$. Now let i be such that $a_{1,i} \neq 0$. The coefficients $a_{1,i}, a_{2,i}, \dots, a_{r,i}$ can be obtained as the first r components of the vector $V^{-1} \det(V)^2 M_0 e_i = \det(V) \cdot \text{Adj}(V) M_0 e_i$. Note that this implies all a_i are in $\mathbb{F}[X, z]$, as all of $V, \text{Adj}(V)$ and M_0 only have polynomial entries. The z -degrees of entries in $\text{Adj}(V)$, and also $\deg_z(\det(V))$, can be bounded by ds^2 , since entries of V have z -degree at most sd due to Item 2, Lemma 10. Furthermore, d bounds the z -degrees of entries of P . So $\text{Adj}(P)$ has entries of z -degrees bounded by sd . Since Q is a matrix with elements in $\{0, -1\}$, this implies the entries of M_0 have degrees bounded by sd . This gives the required bound on the degrees of the a_i s. \square

Let i be given by the above lemma, and fix the vector $u = \det(V)^2 M_0 e_i$. This vector will be the starting point for applying power iteration. To stress, this is an element of $\mathbb{F}[X, z]^s$, since V, M_0 and e_i only contain polynomial entries.

Lemma 13. u can be computed by a multi-output generalized ABP of size $O(s^{1+\gamma_1} + r^{5+\gamma_2} ds)$.

Proof. By Lemma 10, we have an ABP B_1 of size $O(r^{5+\gamma_2} ds)$ computing the polynomial $\det(V)^2$. $M_0 e_i$ is the i th column of $M_0 = \text{Adj}(P)Q$. Note that Q is a projection. Therefore $M_0 e_i$ equals some column of $\text{Adj}(P)$. Each entry of $\text{Adj}(P)$ can be computed by Theorem 3 by a generalized ABP of size $O(s^{\gamma_1})$. This way we obtain a multi-output generalized ABP B_2 computing $M_0 e_i$ of size $O(s^{1+\gamma_1})$. Putting B_1 and B_2 in series gives the required multi-output generalized ABP. \square

5.4 Applying power iteration

Now we are ready to start applying power iteration in order to isolate the single eigenvector v_1 and consequently find the corresponding eigenvalue. We have that $u = a_1 v_1 + a_2 v_2 + \dots + a_r v_r$, for certain $a_i \in \mathbb{F}[X, z]$, as given by Lemma 12. To-

gether with Lemma 10, we can bound for any i , $\deg_z(a_i) + \deg_z(v_i) \leq d^3 s^5 + ds$. For any $\alpha \in \mathbb{F}$, define $P_\alpha = P - \alpha z^d Q$, and $M_\alpha = \text{Adj}(P_\alpha)Q$. Note this definition coincides with previously defined M_0 . We have the following straightforward proposition:

Proposition 6. $\forall \alpha \in \mathbb{F}, P_\alpha v_i = (q_i - \alpha z^d) \cdot Q v_i$.

First we consider what happens when we apply M_α to one of the eigenvectors v_1, v_2, \dots, v_r .

Proposition 7. $\forall i \in [r], e \geq 1, M_\alpha^e v_i = \left(\frac{\det(P_\alpha)}{q_i - \alpha z^d} \right)^e \cdot v_i$, and the entries of $M_\alpha v_i$ are in $\mathbb{F}[X, z]$.

Proof. Since $P_\alpha v_i = (q_i - \alpha z^d) \cdot Q v_i$, we have that $\det(P_\alpha) v_i = \text{Adj}(P_\alpha) P_\alpha v_i = \text{Adj}(P_\alpha) (q_i - \alpha z^d) \cdot Q v_i = (q_i - \alpha z^d) M_\alpha v_i$. Hence, since $q_i - \alpha z^d \neq 0$, we can write $M_\alpha v_i = \frac{\det(P_\alpha)}{q_i - \alpha z^d} v_i$. This proves the case $e = 1$, from which the general case follows trivially. The statement regarding the entries of the vector $M_\alpha v_i$ clear, since the entries of v_i and M_α both lie in $\mathbb{F}[X, z]$. \square

More generally, we have the following statement.

Proposition 8. Given $\alpha_2, \alpha_3, \dots, \alpha_r \in \mathbb{F}$, for all $i \in [r], e \geq 1$ we have that $M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i = \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e \cdot v_i$, and the entries of $M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i$ lie in $\mathbb{F}[X, z]$. Consequently, for any $\ell \in [s]$, we have that

- if $(v_i)_\ell \neq 0$, then the z -degree of the ℓ th component $(M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i)_\ell$ equals $\deg_z((v_i)_\ell) + \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) - \sum_{j=2}^r e \cdot \deg_z(q_i - \alpha_j z^d)$.
- if $(v_i)_\ell \equiv 0$, then $(M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i)_\ell \equiv 0$.

Proof. The very first statement immediately follows from Proposition 7. Let $F_\ell = (M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i)_\ell$. It is clear that if $(v_i)_\ell \equiv 0$, then $F_\ell \equiv 0$. Otherwise, we get that

$$F_\ell \cdot \prod_{j=2}^r (q_i - \alpha_j z^d)^e = \prod_{j=2}^r (\det(P_{\alpha_j})^e \cdot (v_i)_\ell).$$

Think of these as polynomials in z , i.e. elements of $\mathbb{F}[X][z]$. Since $\prod_{j=2}^r (\det(P_{\alpha_j})^e$ and $\prod_{j=2}^r (q_i - \alpha_j z^d)^e$ are both nonzero polynomials, we get that

$$\deg_z(F_\ell) + \deg_z\left(\prod_{j=2}^r (q_i - \alpha_j z^d)^e\right) = \deg_z\left(\prod_{j=2}^r (\det(P_{\alpha_j})^e) + \deg_z((v_i)_\ell).\right)$$

Hence

$$\begin{aligned} \deg_z(F_\ell) + \sum_{j=2}^r e \cdot \deg_z(q_i - \alpha z^d) = \\ \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) + \deg_z((v_i)_\ell). \end{aligned}$$

□

Now consider what happens when we apply $M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e$ to our chosen starting point u . By linearity over \mathbb{G} of M_α , we have that $M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e u = \sum_{i \in [r]} a_i M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i = \sum_{i \in [r]} a_i \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i$, where the last equation follows from Proposition 8. Let

$$g = \sum_{i \in [r]} a_i \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i \quad (3)$$

By Proposition 8, for each $i \in [r]$, the division in $\eta := \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i$ is exact, i.e. η is a vector in $\mathbb{F}[X, z]^s$. Since every $a_i \in \mathbb{F}[X, z]$, we have that $g \in \mathbb{F}[X, z]^s$. Also, Lemma 12 established that $a_1 \neq 0$. Let $R = \deg_z \left(a_1 \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_1 - \alpha_j z^d} \right)^e v_1 \right)$. Recall that the coefficient of z^d in q_1 equals α_1 , and that $\alpha_1, \alpha_2, \dots, \alpha_r$ are distinct elements of \mathbb{F} . Therefore for $j \neq 1$, $q_1 - \alpha_j z^d$ still contains the unique maximum z -degree monomial z^d , i.e. z^d appears in $q_1 - \alpha_j z^d$ with the nonzero coefficient $\alpha_1 - \alpha_j$. Proposition 8 therefore gives us that $R \leq \deg_z(a_1) + \deg_z(v_1) + \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) - e(r-1)d$. For $i > 1$, if $a_i \neq 0$, Let $T_i = \min \deg_z \left(a_i \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i \right)$. Note that for $q_i - \alpha_j z^d$, if $j = i$, we do not have the maximum degree monomial z^d appearing. Proposition 8 therefore gives us that $T_i \geq \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) - \sum_{j=2}^r e \cdot \deg_z(q_i - \alpha_j z^d)$. Straightforward arithmetic gives that $T_i \geq \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) - e(r-1)d + e$.

So for any $e \geq \deg_z(a_1) + \deg_z(v_1) + 1$, we get that $R < T_i$ for every $i \geq 2$ with $a_i \neq 0$. Recall that we observed before that $\deg_z(a_1) + \deg_z(v_1) \leq d^3 s^5 + ds$. We therefore take $e = d^3 s^5 + ds + 1$, and let $\kappa = \sum_{j=2}^r e \cdot \deg_z(\det(P_{\alpha_j})) - e(r-1)d + e - 1$. Note that $\deg_z(\det(P_{\alpha_j})) \leq sd$. Hence $\kappa \leq e(rds + 1) = O(rd^4 s^6)$. We have shown that $[g]_{\leq \kappa} = a_1 \prod_{j=2}^r \left(\frac{\det(P_{\alpha_j})}{q_1 - \alpha_j z^d} \right)^e v_1$.

5.5 Constructing the ABP for the eigenvalue q_1

Iterated matrix multiplication is coded easily with ABPs, which yields the following lemma:

Lemma 14. *The vector $g = M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e u$ can be computed by a generalized multi-output ABP of size $O(r^2 d^3 s^{6+\gamma_1} + r^{5+\gamma_2} ds)$.*

Proof. Starting with the generalized multi-output ABP computing u given by Lemma 13 of size $O(s^{1+\gamma_1} + r^{5+\gamma_2} ds)$, we add stages to compute the required consecutive multiplication by matrices of the form M_α , for $\alpha \in \mathbb{F}$. Each such matrix multiplication can be achieved by adding $O(rs^{1+\gamma_1})$ nodes to the ABP. Namely, $M_\alpha = \text{Adj}(P_\alpha)Q = \text{Adj}(P - \alpha z^d Q)Q$. Since Q is a projection, i.e. M_α consists of r columns selected from $\text{Adj}(P - \alpha z^d Q)$. If we would allow arbitrary polynomials on the wires of ABPs, this means that multiplication by M_α can be realized by one layer that is a bipartite graph with s input nodes and s output nodes with at most rs many edges that are labeled by entries of $\text{Adj}(P - \alpha z^d Q)$. Within the generalized ABP model we can achieve the same, by expanding each such edge into a subprogram computing the appropriate entry of $\text{Adj}(P - \alpha z^d Q)$. By Theorem 3, each entry of $\text{Adj}(P - \alpha z^d Q)$ can be computed by a generalized ABP of size $O(s^{\gamma_1})$. This gives a overall bound of $O(rs^{1+\gamma_1})$ many added nodes to multiply by M_α .

We therefore get that the final ABP for g has size $O(er \cdot rs^{1+\gamma_1} + s^{1+\gamma_1} + r^{5+\gamma_2} ds)$. This gives the required bound stated in the lemma, since $e = d^3 s^5 + ds + 1$ and $r < s$. □

Using Lemma 4, we get:

Corollary 2. *$[g]_{\leq \kappa}$ can be computed by a generalized multi-output ABP of size $O(r^3 d^7 s^{12+\gamma_1} + r^{6+\gamma_2} d^5 s^7)$.*

Let $\tilde{v}_1 = [g]_{\leq \kappa}$. We know that $\tilde{v}_1 \in \mathbb{F}[X, z]$, since $g \in \mathbb{F}[X, z]$. We apply M_0 one more time to obtain the eigenvector corresponding to v_1 . We have that $M_0 \tilde{v}_1 = \left(\frac{\det(P)}{q_1} \right) \tilde{v}_1$. We know that $M_0 \tilde{v}_1 \in \mathbb{F}[X, z]$, since M_0 only contains polynomial entries and $\tilde{v}_1 \in \mathbb{F}[X, z]$. Hence, if ℓ is such that $(v_1)_\ell$ is a nonzero component (which must exist), we get that $\frac{(v_1)_\ell \cdot \det(P)}{(M_0 \tilde{v}_1)_\ell} = q_1$.

The enumerator $(\tilde{v}_1)_\ell \cdot \det(P)$ is computed by series composition of a generalized ABP computing $\det(P)$ with a single output generalized ABP computing $(\tilde{v}_1)_\ell$, which is obtained via Corollary 2. The size can be bounded by $O(r^3 d^7 s^{12+\gamma_1} + r^{6+\gamma_2} d^5 s^7)$. The denominator is obtained by adding one more stage to the ABP from Corollary 2 in order to compute multiplication by M_0 . The size of the resulting single output generalized ABP can be bounded by $O(r^3 d^7 s^{12+\gamma_1} + r^{6+\gamma_2} d^5 s^7)$. Finally, we apply Lemma 2 to perform the exact division. Hence q_1 can be computed by a generalized ABP of size $O((r^3 d^7 s^{12+\gamma_1} + r^{6+\gamma_2} d^5 s^7)^{\gamma_3})$. The ABP for p_1 is obtained by setting $z = 1$ in this ABP. This completes the proof of Main Lemma 8. \square

We can now prove Theorem 1. First we convert the ABP for f to standard form using Lemma 6. This blows up the size to $O(r^2 s)$. Next we apply Lemma 8. The composition of these two operations yields that for the function $\beta(r, d, s)$ from Lemma 5 we can write $\beta(r, d, s) = O((r^3 d^7 (r^2 s)^{12+\gamma_1} + r^{6+\gamma_2} d^5 (r^2 s)^7)^{\gamma_3})$, where γ_1, γ_2 and γ_3 are the absolute constants introduced in Section 2. To be concrete, we get from this that $\beta(r, d, s) = O(r^{396} d^{84} s^{180})$. Hence by Lemma 5, every p_i can be computed by an ABP of size $\beta(r, d, (rs)^{\gamma_3}) = O(r^{2556} d^{84} s^{2160})$. \square

6 Roots of arithmetic circuits and Newton's method

For $f(y) \in \mathbb{R}[y]$ with $f(p) = 0$ for $p \in \mathbb{R}$, recall the update rule for Newton's method $y_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)}$, where f' is the derivative of f . For arithmetic circuits we have the following analogue, where we compute successively better approximations $p_{\leq k}, p_{\leq k+1}, \dots$ to a root $p \in \mathbb{F}[X]$ of $f \in \mathbb{F}[X, y]$.

Lemma 15. *Let $f \in \mathbb{F}[X, y]$ and let $f'(x, y) := \frac{\partial f}{\partial y}$. Let $p \in \mathbb{F}[X]$ be a root of f for y , and assume that $\xi_0 := f'(0, p(0)) \neq 0$. Then $\forall k \geq 1$ it holds that $p_{\leq k+1} = p_{\leq k} - \frac{1}{\xi_0} \cdot f(x, p_{\leq k})_{=k+1}$.*

Proof. Let $r = \deg_y(f)$ and write $f = \sum_{i=0}^r C_i(x) y^i$. So $f'(x, y) = \frac{\partial f}{\partial y} = \sum_{i=1}^r i C_i(x) y^{i-1}$.

The following computation is modulo the ideal I_{k+2} generated by $x_1^{k+2}, x_2^{k+2}, \dots, x_n^{k+2}$, i.e. we identify any polynomials g and h if $[g]_{\leq k+1} = [h]_{\leq k+1}$.

$$\begin{aligned} 0 &\equiv f(x, p) \\ &\equiv f(x, p_{\leq k} + p_{=k+1}) \\ &\equiv \sum_{i=0}^r C_i(x) (p_{\leq k} + p_{=k+1})^i \\ &\equiv C_0(x) + \\ &\quad \sum_{i=1}^r C_i(x) ((p_{\leq k})^i + i \cdot (p_{\leq k})^{i-1} \cdot p_{=k+1}) \\ &\equiv \sum_{i=0}^r C_i(x) (p_{\leq k})^i + \\ &\quad p_{=k+1} \cdot \sum_{i=1}^r i \cdot C_i(x) (p_{\leq k})^{i-1} \\ &\equiv f(x, p_{\leq k}) + p_{=k+1} \cdot f'(x, p_{\leq k}) \\ &\equiv f(x, p_{\leq k}) + p_{=k+1} \cdot f'(x, p_{\leq k})_{=0}. \end{aligned}$$

Note that $f'(x, p_{\leq k})_{=0} = f'(0, p_{\leq k}(0)) = f'(0, p(0)) = \xi_0$. We get that without going modulo I_{k+2} , the following equation is satisfied: $0 = f(x, p_{\leq k})_{=k+1} + p_{=k+1} \cdot \xi_0$. This implies the statement of the lemma. \square

6.1 Proof of Theorem 2

Let $f'(x, y) := \frac{\partial f}{\partial y}$. In case $f'(0, p(0)) \neq 0$, we can construct an arithmetic circuit for p by repeatedly applying Lemma 15. We compute the components of p separately, starting with p_0 and p_1 , which we can easily compute within size $O(s)$. To compute $p_{=k+1}$, provided we have p_0, p_1, \dots, p_k computed at gates somewhere already, we use a copy of a circuit Φ that computes the homogeneous components of f up to degree $k+1 \leq \deg(p)$. This is a circuit for which, similar to the proof of Lemma 3, each node is split into $k+1$ nodes computing homogeneous components. Let v_0, v_1, \dots, v_{k+1} be the gates in Φ corresponding to the output gate of the original circuit, i.e. $f_0, f_1, \dots, f_{=k+1}$ are computed at these gates. We can bound the size of Φ by $O(\mathcal{M}(k+1)s)$, provided we use a gadget of size $\mathcal{M}(k+1)$ that computes the coefficient map of polynomial multiplication, in order to deal with multiplication. Note that having p_0, p_1, \dots, p_k computed separately at gates is exactly the right format for feeding $p_{\leq k}$ into Φ for the variable y . A straightforward structural induction proves that after rewiring, for every $0 \leq i \leq k+1$, the gate v_i computes $f(x, p_{\leq k})_i$. Lemma 15 tells us that after rescaling the output of the gate v_{k+1} by a factor $-1/\xi_0$, we have obtained $p_{=k+1}$. We repeat the previously described construction for k up to degree $\deg(p)$. This way, we obtain a circuit for p of size $O(\mathcal{M}(\deg(p)) \cdot \deg(p) \cdot s)$.

If $f'(0, p(0)) = 0$, then we can reduce to the above case as follows. Write $f = \sum_{i=0}^r C_i(x)y^i$ with $C_r(x) \neq 0$. Let $f^i(x, y) = \frac{\partial^i f}{\partial y^i}$. Then $f^r(x, y) = r! \cdot C_r(x)$. Since the characteristic of \mathbb{F} is zero, $r! \neq 0$, so $f^r(x, p) \neq 0$. We have in this case that $f^0(x, p) \equiv 0$. Let i be the smallest integer for which $f^i(x, p) \neq 0$. Then $0 < i \leq r$, and $f^{i-1}(x, p(x)) \equiv 0$. Due to Lemma 1, $f = (y - p)^m h$, for some polynomial h not divisible by $y - p$. By repeatedly computing partial derivatives one easily observes that the number i equals the multiplicity m of the root p in f .

We have that there exists $x_0 \in \mathbb{F}$ such that $f^i(x_0, p(x_0)) \neq 0$. Let $g(x, y) = f^{i-1}(x + x_0, y)$, and let $q = p(x + x_0)$. By Theorem 4, one gets that g is computable by a circuit of size $O(\mathcal{M}(m)s)$. Let $g' = \frac{\partial g}{\partial y}$. Then $g'(x, y) = f^i(x + x_0, y)$. The polynomial g is not identically zero, and $g(x, q(x)) = f^{i-1}(x + x_0, p(x + x_0)) \equiv 0$, and furthermore $g'(0, q(0)) = f^i(x_0, p(x_0)) \neq 0$.

Now one proceeds as in the first case, to get a circuit for q of size $O(\mathcal{M}(m)\mathcal{M}(\deg(p)) \cdot \deg(p) \cdot s)$, from which one obtains a circuit for p of size $O(\mathcal{M}(m)\mathcal{M}(\deg(p)) \cdot \deg(p) \cdot s)$. \square

Corollary 3. *The class is closed under taking roots.*

References

- [1] Erich Kaltofen. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.
- [2] P. Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004.
- [3] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity testing means proving circuit lower bounds. *Computational Complexity*, 13(1–2):1–44, 2004.
- [4] Z. Dvir, A. Shpilka, and A. Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. In *Proceedings of the 40th Annual STOC*, pages 741–748, 2008.
- [5] M. Jansen. Weakening assumptions for deterministic subexponential time non-singular matrix completion. In *27th International Symposium on Theoretical Aspects of Computer Science (STACS 2010)*, volume 5 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 465–476, 2010.
- [6] D.G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.
- [7] E. Kaltofen and P. Koiran. Expressing a fraction of two determinants as a determinant. In *Proceedings, The 19th International Symposium on Symbolic and Algebraic and Computation (ISSAC)*, pages 141–146, 2008.
- [8] M. Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago Journal of Theoretical Computer Science*, 1997(Article 5), 1997.
- [9] N. Alon. Combinatorial nullstellensatz. *Combinatorics, Probability and Computing*, 8(1–2):7–29, 1999.
- [10] E. Kaltofen and M. Singer. Size-efficient parallel algebraic circuits for partial derivatives. In V. Shirkov, V.A. Rostovtsev, and V.P. Gerdt, editors, *Proceedings, IV International Conference on Computer Algebra in Physical Research*, pages 133–145. World Scientific, 1991.
- [11] L. Valiant. Completeness classes in algebra. In *Proc. 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261, 1979.
- [12] A. Shpilka and A. Wigderson. Depth-3 arithmetic formulae over fields of characteristic zero. *Journal of Computational Complexity*, 10(1):1–27, 2001.