Electronic Colloquium on Computational Complexity, Report No. 118 (2010)

# Extracting Roots of Arithmetic Circuits by Adapting Numerical Methods

Maurice Jansen [*]

July 27, 2010

**Abstract**

For two polynomials $f \in \mathbb{F}[x_1, x_2, \ldots, x_n, y]$ and $p \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, we say that $p$ is a *root* of $f$, if $f(x_1, x_2, \ldots, x_n, p) \equiv 0$. We study the relation between the arithmetic circuit sizes of $f$ and $p$ for general circuits and *skew* circuits. Arithmetic skew circuits are defined by restricting every multiplication gate to have at least one of its inputs equal to a variable or a field constant. They were introduced by Toda [1], who showed they capture the complexity of the determinant polynomial.

We address the following fundamental question: suppose the polynomial $f$ can be computed by a skew circuits of size $s$. Is the skew circuit 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 $poly(s, deg(p), m)$, 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 [2] and Bürgisser [3].

In this paper, we study the above question for skew circuits for the canonical case where $f$ is assumed to factor as

$$f = p_0 \cdot (p_1 - y)(p_2 - y) \ldots (p_r - y),$$

for $p_0, p_1, \ldots, p_r \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ with $p_0 \not\equiv 0$, and where $p_1, p_2, \ldots, p_r$ 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 a skew circuit of size polynomial in $s$. This demonstrates an important special case where the answer to the above mentioned question is affirmative. Prior to this paper, no method was known to provide a $poly(s)$ bound for this main scenario.

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 $\mathbb{F}(x_1, x_2, \ldots, x_n)$ 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 a skew circuit, since skew circuits can efficiently compute iterated matrix multiplication.

A novel aspect of this work is that 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 $poly(s, deg(p), m)$ 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 Ref. [2, 3], we develop a discrete analogue of Newton's Method.

## 1 Introduction

For an arithmetic circuit class $\mathcal{C}$, whether $\mathcal{C}$ is closed under taking roots is a fundamental question, and 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

[*] Institute for Theoretical Computer Science, Tsinghua University. Email: maurice.julien.jansen@gmail.com. This work was supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grant 2007CB807900,2007CB807901.

ISSN 1433-8092

derandomization of *polynomial identity testing* (PIT) for the class $\mathcal{C}$. For the latter well-known problem one is given an arithmetic circuit $\Phi$, and the problem is to decide whether the polynomial computed by $\Phi$ is identical to the zero polynomial or not. Due to a result independently obtained by Kaltofen [2] and Bürgisser [3], 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 [4] 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 considering the question whether PIT for $\mathcal{C}$ can be achieved deterministically under any weaker assumptions. When using the framework of Ref. [4], 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 [5] and Jansen [6].

In Ref. [5] 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 skew circuits a root extraction lemma is proved in Ref. [6], 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 skew circuits can be solved deterministically in subexponential time, assuming some explicit polynomial is hard for skew circuits. In this paper we make progress towards showing that the arithmetic circuit class $\mathrm{VP}_{skew}$ of polynomial families computable by *poly* size skew circuits is closed under taking roots. The latter statement, *if true*, would yield[1] a deterministic subexponential time PIT algorithm for $\mathrm{VP}_{skew}$, under the assumption that there exists some explicit family of polynomials that requires skew circuits of super-polynomial size.

Already implicit in Ref. [5, 6] 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 may serve as a conceptual simplification in Ref. [4], in the sense that calling upon the more involved works Ref. [2, 3] is avoided. For skew circuits 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 skew circuits can efficiently compute matrix multiplication, we avoid the explosion in skew circuit 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 $\lambda_1 > \lambda_2 > \ldots > \lambda_s > 0$ and a corresponding independent set of unit eigenvectors $v_1, v_2, \ldots, v_s$, a well-known heuristic for finding an approximation to the largest eigenvalue $\lambda_1$ is to apply power iteration. Here, starting with some vector $u$ that is typically selected at random, writing $u$ in the eigenbasis as $u = a_1 v_1 + a_2 v_2 + \ldots + a_s v_s$, for certain scalars $a_i$, one applies a large power of $M$ to $u$ to obtain $M^e u = a_1 \lambda_1^e v_1 + a_2 \lambda^e v_2 + \ldots + a_s \lambda^e v_s$. 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 $e \to \infty$. Once an approximation $\tilde{v_1}$ to $v_1$ is obtained, one may approximate $\lambda_1$ 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 skew circuits for roots of skew circuits. 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 relatively unexplored direction.

[1] Already 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 skew circuit size over *all* fields.

## 1.1 Results

For the rest of the paper, we prefer to work with the technically more convenient *algebraic branching program*[2] model. An algebraic branching program (ABP) is given by a layered directed acyclic graph with source $\sigma$ and sink $\tau$, whose edges are labeled by variables or field constants. It computes the sum of weights of all paths from $\sigma$ to $\tau$, 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. As an easy exercise, given a polynomial $f$, one can switch between skew circuits and ABPs for $f$ with at most a quadratic blow-up in the size. Our main result is the following theorem:

**Theorem 1.** *Let $\mathbb{F}$ be a field of characteristic zero. Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n, y]$ be a nonzero polynomial that can be computed by an ABP of size $s$. Suppose $f$ factors as*

$$f = p_0(p_1 - y)(p_2 - y)\ldots(p_r - y),$$

*where $\{p_0, p_1, p_2, \ldots, p_r\} \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ and $p_1, p_2, \ldots, p_r$ are pairwise distinct. Then every $p_i$ has an ABP of size at most polynomial* [3] *in $d, r$ and $s$, where $d = \max_{i\in[r], p_i\not\equiv 0} deg(p_i)$.*

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 example, if $p_1, p_2, \ldots, p_r$ are all nonzero, then $f$ contains a term of degree at least $deg(p_1 p_2 \ldots p_r) \geq d$. Hence the theorem implies that for every root we have an ABP of size $poly(s)$. For comparison, Lemma 2.10 of Ref. [6] yields[4] 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 $\mathbb{F}[z]$, given the coefficients of $g$ and $h$ as input variables. By a result of Cantor and Kaltofen [7], 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 $f \in \mathbb{F}[x_1, x_2, \ldots, x_n, y]$ 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. $f(x_1, x_2, \ldots, x_n, p) \equiv 0$ 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 $\mathbb{F}[x_1, x_2, \ldots, x_n, y]$. Using Ref. [2, 3] to obtain arithmetic circuits for the factor $p - y$, as done in Ref. [4], 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 and Techniques

To establish Theorem 1 we more or less will follow the following program:

1. Reduction to 'nice' polynomials.

   The first step in the proof is to show that the general 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 $p_1(0), p_2(0), \ldots, p_r(0)$ 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] See Section 2 for a formal definition.

[3] In 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 $O(r^{396}d^{84}s^{180})$.

[4] Note that this lemma is stated for skew circuits, but inspection of the proof shows that that the quadratic blow-up of $s$ can be avoided.

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 $q_i = z^d p_i(\frac{x_1}{z}, \frac{x_2}{z}, \ldots, \frac{x_n}{z})$ and $d = max_i deg(p_i)$. For $\alpha \in \mathbb{F}$, we consider *shifting* by $\alpha z^d$, 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 $f_{\alpha_2}, f_{\alpha_3}, \ldots, f_{\alpha_r}$, since all $\alpha_i$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 $f_{\alpha_2}, f_{\alpha_3}, \ldots, f_{\alpha_r}$.

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 $P_\alpha$, whose entries are products of $x_i$ variables and powers of $z$. Let $M_\alpha = Adj(P_\alpha)Q$, where $Adj(P_\alpha)$ denotes the *adjugate* of $P_\alpha$, i.e. $P_\alpha^{-1} = Adj(P_\alpha)/\det(P_\alpha)$. 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 $M_\alpha$, 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 $\lambda_i$, whose entries are *polynomials.* We take the matrix $V = [v_1, v_2, \ldots, v_r, e_{i_1}, \ldots, e_{i_{s-r}}]$, where we extend with some standard basis vectors $e_{i_j}$ to make $V$ nonsingular. For an arbitrary point $u \in Range(M_0)$, writing $u = a_1 v_1 + \ldots + a_r v_r$, means we have to apply $V^{-1} = Adj(V)/det(V)$ 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 $v_1, v_2, \ldots, v_r$ 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 $S_r^j(x_1, \ldots, x_r) = \sum_{I\subset[r]} \prod_{i\in I} x_i$, due to the fundamental theorem on symmetric polynomials. Eventhough we cannot directly compute the $q_i$s, $S_r^j(q_1, \ldots, q_r)$ equals the coefficient of $(-y)^{r-j}$ in $f'$ (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 $M_{\alpha_r}^e \ldots M_{\alpha_3}^e M_{\alpha_2}^e u$, where $e$ is some appropriately selected large integer. The next crucial part of the proof is to show that when changing $\alpha$, only the eigenvalues of $M_\alpha$ 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:

$$u' := M_{\alpha_r}^e \ldots M_{\alpha_3}^e M_{\alpha_2}^e u \quad = \quad \sum_{i\in[r]} a_i \prod_{j=2}^{r} \left( \frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i.$$

   For the above expression 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 $M_\alpha$ 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 [8] to perform the (exact) division of two ABPs. Finally, the ABP for $p_1$ is obtained by setting $z = 1$ in the ABP for $q_1$.

### 1.3 Structure of the Rest of the Paper

Section 2 is a preliminaries section. In Section 3 we prove the reduction to so-called nice polynomials. In Section 4 we develop the needed notions regarding standard form ABPs, homogenization and completeness of the determinant. In Section 5 we prove our main root extraction lemma for nice polynomials. Section 6 is a short section in which we use the main lemma to prove Theorem 1. Finally, we prove Theorem 2 in Section 7, where we consider Newton's Method and root extraction for unrestricted arithmetic circuits.

## 2 Preliminaries

Let $\mathbb{F}$ be a field of characteristic zero. Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of indeterminates. Let $\mathbb{G}$ denote the field of rational functions $\mathbb{F}(X, z)$. For a polynomial $f \in \mathbb{F}[X, y]$ 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 $\text{Vandet}(x_1, x_2, \ldots, x_n) = \prod_{1 \le i < j \le n}(x_i - x_j)$. 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 $deg(v) = \max_{j, v_j \not\equiv 0} deg(v_j)$ and $deg_{x_i}(v) = \max_{j, v_j \not\equiv 0} deg_{x_i}(v_j)$. 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 $f_{\le i}$, $f_{\ge i}$, $[f]_{\le i}$ and $[f]_{\ge i}$. We will also use this notation for vectors of polynomials. For example, $(f, g, h)_{\le i} = (f_{\le i}, g_{\le i}, h_{\le i})$. At a few occasions we will also use the notation $f_{\le_z i}$, 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 $mindeg(v) = \min_{j, v_j \not\equiv 0} mindeg(v_j)$ and $mindeg_{x_i}(v) = \min_{j, v_j \not\equiv 0} mindeg_{x_i}(v_j)$. Given an integer $d \ge 0$ and a variable $z$, for a polynomial $f$ of degree $\le d$, the *homogenization of $f$ to degree $d$ using the variable $z$*, is the polynomial $z^d f(\frac{x_1}{z}, \frac{x_2}{z}, \ldots, \frac{x_n}{z})$.

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:

**Proposition 1.** *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 $M$, $M Adj(M) = Adj(M) M = \det(M) I$, where $I$ denotes the identity matrix.

### 2.1 Arithmetic Circuits and Algebraic Branching Programs

An *arithmetic circuit* $\Psi$ 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 $\Psi$ we have associated a polynomial in $\mathbb{F}[X]$ computed by $g$, which is defined in the standard manner. The output of $\Psi$ is the polynomial computed by some designed output gate. For the size of $\Psi$ 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 $deg(f_n) \le p(n)$ and $s(f_n) \le p(n)$.

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* $L_0, L_1, \ldots, L_\ell$, where $L_0 = \sigma$ and $L_\ell = \tau$. Vertices $\sigma$ and $\tau$ are called the source and sink of $\Phi$, 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 $\Phi$. The weight function $w : E \to X \cup \mathbb{F}$ 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 $\Phi$ computes the polynomial $\hat{\Phi} := \sum_{p \in P_{\sigma,\tau}} w(p)$. The size of $\Phi$ is defined to be $|V|$. For nodes $v$ and $w$ in $\Phi$, $\Phi_{v,w}$ denotes the subprogram of $\Phi$ 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 $\mathcal{W}$. For example, an edge is allowed to have weight $z^{10}x_1$, or just $z^5$, or any element from $X \cup \mathbb{F}$ as before. The output of such an ABP is an element of $\mathbb{F}[X, z]$, and is defined in the obvious way.

We will also consider "multi-output" ABPs. In this case the last layer of the $A$ consists of several sink nodes $\tau_1, \tau_2, \ldots, \tau_m$. The output of the ABP is given by the tuple of polynomials $(f_1, f_2, \ldots, f_m)$ computed by the subprograms $A_{\sigma,\tau_1}, A_{\sigma,\tau_2}, \ldots, A_{\sigma,\tau_m}$. Mainly, we use this feature for conveniently constructing larger single-output ABPs. For example, if we want to apply some $m' \times m$ matrix $M$ to the vector $(f_1, f_2, \ldots, f_m)^T$, note that this is easily done by adding one more layer with edge labels corresponding to entries of $M$. In particular, for $m' = 1$ with entries of $M$ in $\mathbb{F}$, this produces a single output ABP computing a linear combination of $f_1, f_2, \ldots, f_m$.

One main reason for using ABPs rather than skew circuits, is that they are more convenient when doing substitution. It is easily seen that if $g(x_1, x_2, \ldots, x_n)$ can be computed by an ABP $A_g$ of size $s_g$ and $f(x_1, x_2, \ldots, x_n)$ 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 $A_f$. Indeed, simply 'expand' every edge labeled $x_i$ in $A_f$ into a subprogram $A_g$. This can be done while keeping the program leveled with the final size being $O(e_f s_g)$.

We use the following result by Mahajan and Vinay:

**Theorem 3** (See Theorem 2 in [9]). *The determinant of an $n \times n$ matrix can be computed by an ABP of size $O(n^3)$ with $O(n^5)$ many edges.*

For the analysis, we let $\gamma_1$ and $\gamma_2$ be universal constants such that the determinant of an $n \times n$ matrix can be computed by an ABP with $\gamma_1$ many nodes and $\gamma_2$ many edges. By the above we can take $\gamma_1 = 3$ and $\gamma_2 = 5$. Also, we define the universal constant $\gamma_3$ to be the best possible constant for the exponent of $s$ in the following lemma (so $\gamma_3 \leq 12$):

**Lemma 2** (From Lemma 1 and Lemma 2 in [8][5]). *Suppose $|\mathbb{F}|$ is infinite. Let $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ 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 $O(s^{12})$.*

The following two lemma's are proved by the well-known trick of 'splitting' nodes in order to keep track of degree components. For Lemma 3 this is done to keep track of total degree, after which the appropriate powers of $z$ can be attached to construct the final ABP. For Lemma 4 the trick is applied to keep track of components with different degree in $z$, which allows the final ABP to be coded easily. For example, an edge $(v, w)$ with label $x_i \cdot z^k$ induces edges $(v_i, w_{i+k})$ with label $x_i$, for all $i + k \leq d$, where $v$ and are split into $v_0, v_1, \ldots, v_d$ and $w_0, w_1, \ldots, w_d$, respectively.

**Lemma 3.** *Let $d \geq 0$ be an integer and $z$ a new variable. Let $\Phi$ be an ABP of size $s$ computing the polynomial $f \in \mathbb{F}[X]$ of degree at most $d$. Then there exist an ABP $\Psi$ 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 $\Phi$ 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 $\Psi$ of size $O(ds)$ and $O(de)$ many edges computing $[f]_{\leq_z d}$. A similar statement holds for computing $[f]_{=_z d}$.*

[5] We get an extra quadratic blow-up of $s$, since the DAG we use for ABPs must be leveled, as opposed to the DAG constructed in Lemma 2 of Ref. [8].

We also need a lemma by Alon:

**Lemma 5** (Lemma 2.1 in [10]). *Let $f \in \mathbb{F}[X]$ be a nonzero polynomial such that the degree of $f$ in $x_i$ is bounded by $r_i$, and let $S_i \subseteq \mathbb{F}$ be of size at least $r_i + 1$, for all $i \in [n]$. Then there exists $(s_1, s_2, \ldots, s_n) \in S_1 \times S_2 \times \ldots \times S_n$ with $f(s_1, s_2, \ldots, s_n) \neq 0$.*

Finally, we use a result by Kaltofen and Singer for computing partial derivatives. For a polynomial $f \in \mathbb{F}[X, y]$ and integer $k \geq 0$, $\frac{\partial^k f}{\partial^k y}$ denotes the $k$th order formal partial derivate of $f$ w.r.t. the variable $y$. In the following, $\mathcal{M}(d)$ is the previously introduced function $\mathcal{M}$, that gives an upper bound on the arithmetic circuit size for multiplication of two univariate polynomials of degree $d$ over $\mathbb{F}$, given their coefficients as input variables.

**Theorem 4** (Theorem 3.1 in [11]). *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

**Definition 1.** *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 $\{p_1, p_2, \ldots, p_r\}$ 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.*

**Definition 2.** *A polynomial $f \in \mathbb{F}[X, y]$ is called nice, if $f$ factors as $f = (p_1 - y)(p_2 - y) \ldots (p_r - y)$, where $\{p_1, p_2, \ldots, p_r\} \subset \mathbb{F}[X]$ 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 $\beta$, where $s$ denotes the ABP size of $f$ and $d = \max_i deg(p_i)$. We reduce the more general case of Main Theorem 1 to root extraction for nice polynomials as follows:

**Lemma 6.** *Suppose $|\mathbb{F}|$ is infinite. Suppose $f \in \mathbb{F}[X, y]$ factors as $f = p_0(p_1 - y)(p_2 - y) \ldots (p_r - y)$, where $\{p_0, p_1, p_2, \ldots, p_r\} \subset \mathbb{F}[X]$ and $p_1, p_2, \ldots, p_r$ are pairwise distinct. Suppose $f$ 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 $\gamma_3$ 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 $p_1, p_2, \ldots, p_j$ are constant. Since $j \leq r \leq s$, it is easily seen we have an ABP of size $O(rs)$ that computes $p_0(p_1 - y)(p_2 - y) \ldots (p_j - y)$. Now use Lemma 2 to obtain an ABP of size $s = O(r^{\gamma_3} s^{\gamma_3})$ computing $\tilde{f} = (p_{j+1} - y)(p_{j+2} - y) \ldots (p_r - y)$. 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 $f = (p_1 - y)(p_2 - y) \ldots (p_r - y)$, where $p_1, p_2, \ldots, p_r$ 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 $f' := f(x_1 + a_1, x_2 + a_2, \ldots, x_n + a_n, y)$. We have that

$$\begin{aligned} f' &= \prod_{i=1}^{r} (p_i(x_1 + a_1, x_2 + a_2, \ldots, x_n + a_n) - y) \\ &= \prod_{i=1}^{r} (p_i(x_1 + a_1, x_2 + a_2, \ldots, x_n + a_n)_{\geq 1} + p_i(a_1, a_2, \ldots, a_n) - y). \end{aligned}$$

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 $p_i(x_1 + a_1, x_2 + a_2, \ldots, x_n + a_n)$ 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 $p_i(x_1, x_2, \ldots, x_n)$ of size $O(\beta(r, d, O(r^{\gamma_3} s^{\gamma_3}))$. □

# 4 Standard Form ABPs, Valiant Matrices and Homogenizations

**Definition 3.** *Let $f \in \mathbb{F}[X, y]$ be a polynomial whose degree in $y$ equals $r$, and write $f = \sum_{i=0}^{r} C_r(x)y^r$. We say an ABP $\Phi$ with source $\sigma$ and sink $\tau$ computing $f$ is in standard form, if it has the following structure:*

- *There is a set of distinct nodes $\{b_0, b_1, \ldots, b_r\}$, such that for each $i \in \{0, 1, \ldots, r\}$, there is an edge from the source $\sigma$ to $b_i$ with label 1. These are the only edges adjacent to the source.*
- *There are distinct nodes $c_0, c_1, \ldots, c_r$. The subprograms in the set $\{\Phi_{b_i,c_i} : i \in [r]\}$ are disjoint as graphs. For every $i \in \{0, 1, \ldots, r\}$, the subprogram $\Phi_{\sigma,c_i}$ computes $C_i(x)$.*
- *There is a path $c_r = a_0, a_1, \ldots, a_{r-1}, a_r = \tau$, where each edge $(a_i, a_{i+1})$ is labeled with the variable $y$. These are the only occurrences of $y$ variables in $\Phi$.*
- *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 $\sigma$ to $\tau$ is even.*

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

The standard form is exemplified in Figure 1, where edges drawn without labels carry the field constant 1 by convention. We remark that for a standard form ABP $\Phi_{b_i,c_i}$ and $\Phi_{\sigma,c_i}$ compute the same polynomial. For generalized standard form ABPs the computed polynomials differ by a factor of $w(\sigma, b_i)$.

**Lemma 7.** *Let $f \in \mathbb{F}[X, y]$ be computed by an ABP $\Phi$ of size $s$, and let $r = deg_y(f)$. Then $f$ can be computed by an ABP $\Psi$ in standard from of size $O(sr^2)$. This means in particular that the variable $y$ appears exactly $r$ times on an edge in $\Psi$.*

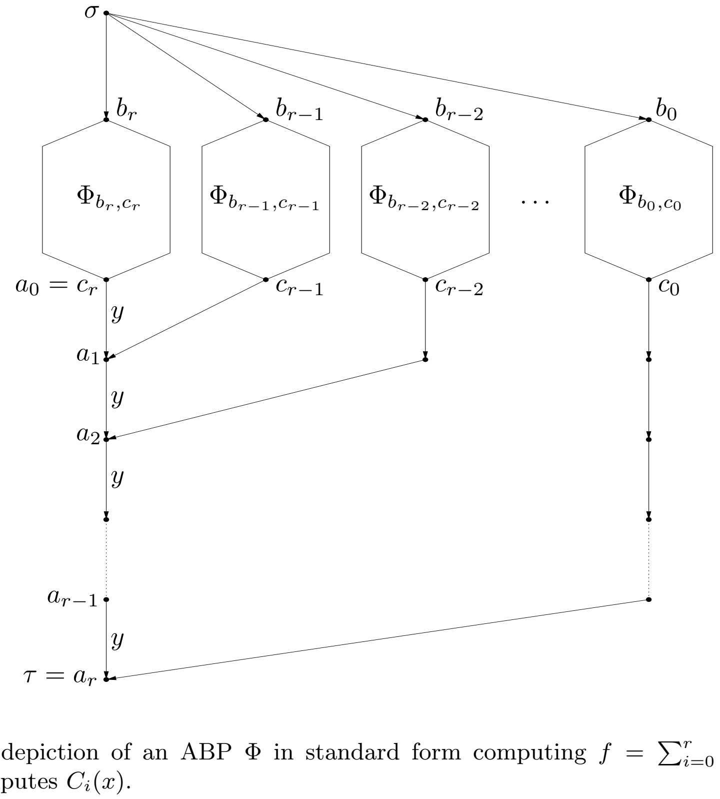

Figure 1: Schematic depiction of an ABP $\Phi$ in standard form computing $f = \sum_{i=0}^{r} C_r(x)y^r$. For each $i$, the subprogram $\Phi_{b_i,c_i}$ computes $C_i(x)$.

*Proof.* Write $f = \sum_{i=0}^{r} f_r(x_1, x_2, \dots, x_n)y^i$. Each $f_i$ can be computed by an ABP of size $O(sr)$ by Lemma 4. Given that we have such ABPs, the final ABP $\Psi$ of size $O(sr^2)$ is obtained in a straightforward manner. □

Given an ABP $\Phi$ 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 [12]. Namely, thinking of $\Phi$ as a graph, one adds a loop back from $\tau$ to $\sigma$ with label 1, and one puts a self loop on all nodes other than $\sigma$ and $\tau$ with label 1. Let $M(\Phi)$ be the[6] adjacency matrix of the weighted graph obtained this way, which we call the *Valiant matrix associated to* $\Phi$. Assuming wlog. that the length of every path from $\sigma$ to $\tau$ in $\Phi$ is even[7], then $\det(M(\phi)) = f$. To make this clear, let us introduce the following notion:

**Definition 4.** *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 $sgn(C)$ to be $-1$ if $C$ is of even length, and $1$ otherwise. For the cycle cover $C$, define $sgn(C) = \prod_i sgn(C_i)$.*

If $M_G$ is the adjacency matrix of a weighted graph $G$, observe that $\det(M_G)$ equals $\sum_C sgn(C)w(C)$, where the sum is over all cycle covers $C$ of $G$. In $M(\Phi)$ cycle covers correspond exactly to paths from the source to the sink in $\Phi$. Namely, each path from $\sigma$ to $\tau$ is completed with the loop from $\tau$ back to $\sigma$, and self loops are added for all nodes not in this cycle. Note every cycle cover has sign 1, since the length of any $\sigma, \tau$-path is even. We conclude that $\det(M(\Phi)) = f$.

Rows and columns of $M(\Phi)$ correspond to nodes in $\Phi$. In our notation, we will use variable names of nodes of $\Phi$ to index the matrix $M(\Phi)$, and also do this for $s$-vectors operated on. For example, for the standard form ABP $\Phi$ for Definition 3, the entry $M(\Phi)_{a_0 a_1}$ equals $y$. More generally, if we start with a generalized ABP $\Phi$, we define in a completely analogous fashion the associated *generalized* Valiant matrix $M(\Phi)$. In this case, $M(\Phi)$ contains elements from $\mathcal{W}$, since whatever weight an edge in $\Phi$ carries goes into the corresponding entry of $M(\Phi)$.

**Definition 5.** *Let $z$ be a variable. Given the Valiant matrix $M(\Phi)$ associated to an ABP $\Phi$, 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 $\Phi$, its d-homogenization $\overline{\Phi}$ is the generalized ABP obtained by performing the above replacement operations (1,2,3) for every edge label.*

Note that $M(\overline{\Phi})$ and $\overline{M(\Phi)}$ are different matrices. The following proposition shows how they relate:

**Proposition 2.** *Let $\Phi$ be an ABP of size $s$ with source $\sigma$ and sink $\tau$. For the case of d-homogenization, we have the following two properties:*

1. *$M(\overline{\Phi})$ and $\overline{M(\Phi)}$ only differ for nonzero entries on the diagonal and the 'loopback' entry on row $\tau$, column $\sigma$. 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 $\ell$ equals the number of layers of $\Phi$.*

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

[7] Note that since ABPs are leveled all paths from $\sigma$ to $\tau$ are of the same length. Also note that if this length is odd, then $\det(M(\phi)) = -f$.

*Proof.* The first property clearly holds. For the second property, let $G$ be the weighted graph with adjacency matrix $M(\overline{\Phi})$). Then $\det(M(\overline{\Phi}))$ equals $\sum_C sgn(C)w(C)$, where the sum is over all cycle covers $C$ of $G$. These cycle covers correspond to paths from $\sigma$ to $\tau$. For each such path, there is added the loop-back edge from $\tau$ to $\sigma$ to complete the 'big cycle'. In addition $s-(\ell+1)$ self-loops are added to cover all vertices not covered in the big loop. In the computation of $\det(M(\overline{\Phi}))$ these edges carry the label 1. By the first property, we can think of $\det(\overline{M(\Phi)})$ as a sum of cycle covers in the graph $G$, but where the weights of the loopback entry and the $s-(\ell+1)$ self-loops have been changed from 1 to $z^d$. This results in an extra factor of $z^{(s-\ell)d}$ per cycle cover, and yields the second property. □

**Proposition 3.** *Suppose $\Phi$ is an ABP of size $s$ computes $f \in \mathbb{F}[X, y]$, where $f$ factors as*

$$f = (p_1 - y)(p_2 - y)\dots(p_r - y).$$

*Let $d = \max_i deg(p_i)$. Then for d-homogenization $\overline{M(\Phi)}$ we have that*

$$\det(\overline{M(\Phi)}) = z^{d(s-r)}(q_1 - y)(q_2 - y)\dots(q_r - y),$$

*where $\forall i \in [r]$, $q_i = z^d p_i(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$. In other words, $q_i$ is the homogenization of the polynomial $p_i$ to degree $d$.*

*Proof.*

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

□

We have the following easily proved decomposition Lemma for Valiant matrices associated to ABPs in standard form:

**Lemma 8.** *Let $f \in \mathbb{F}[X, y]$ be computed by a standard form ABP $\Phi$ of size $s$ an let $r = deg_y(f)$. Suppose that $f(x_1, x_2, \dots, x_n, 0) \not\equiv 0$. Then the associated Valiant matrix $M(\Phi)$ can be written as*

$$M(\Phi) = A(x_1, x_2, \dots, x_n) - yB,$$

*where*

1. *$A$ and $B$ are square matrices of order $s$.*
2. *$B$ is a matrix whose entries are taken from the set $\{0, -1\}$. In each row/column there is at most one -1, and the number of -1s in $B$ equals $r$.*
3. *$A$ has entries taken from the set $\mathbb{F} \cup \{x_1, x_2, \dots, x_n\}$.*
4. *$A$ is invertible, if we consider it over the field of rational functions $\mathbb{F}(X)$.*

*Proof.* The properties of the lemma follow easily by taking $B$ to have -1 precisely in the places where $M(\Phi)$ has $y$'s and zeros everywhere else. In particular, for the last property, since $\det(A) = \det(M(\Phi)_{|y=0}) = \det(M(\Phi))_{|y=0} = f(x_1, x_2, \dots, x_n, 0) \not\equiv 0$, we conclude that $A$ is invertible over $\mathbb{F}(X)$. □

## 4.1 A Closed Form for Eigenvectors Related to the Valiant Matrix

**Proposition 4.** *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 $Mv = 0$.*

*Proof.* For any $k \in [m]$, using Proposition 1, we get that $(Mv)_k = \sum_{\ell \in [m]} M_{k\ell} v_\ell = \sum_{\ell \in [m]} M_{k\ell} \det(M[i,\ell]) = \sum_{\ell \in [m]} M_{k\ell} (-1)^{i+\ell} \det(M[[i,\ell]])$. If $k = i$, this equals the Laplace expansion for $\det(M)$ along row $i$, so $(Mv)_k = \det(M) = 0$. If $k \neq i$, this equals the Laplace expansion of the matrix $M'$ obtained by replacing the $i$th row with the $k$th row of $M$. Since $M'$ contains the same row twice, $\det(M') = 0$, which implies that also in this case $(Mv)_k = 0$. □

The following lemma provides us with a closed form solution $v$ for the system of linear equations $Nv = 0$, with $N = M(\Phi)_{|y=q}$, derived from the associated Valiant matrix $M(\Phi)$ of a generalized *standard form* ABP $\Phi$ computing some polynomial $f$, where $q$ is a root of $f$. If $q$ is a nonzero root of $f$, and expressing $M(\Phi) = A - yB$ as in Lemma 8, then it is easily seen that $Adj(A)Bv = \frac{\det(A)}{q} \cdot v$. In other words, $v$ is an eigenvector of $Adj(A)B$ corresponding to the eigenvalue $\frac{\det(A)}{q}$.

**Lemma 9.** *Let $\Phi$ be a generalized ABP of size $s$ in standard form computing the polynomial $f = \sum_{i=0}^{r} C_i(X,z)y^r \in \mathbb{F}[X,z,y]$ of degree $r$ in $y$. Let $d$ be a bound on the $z$-degree of edge labels in $\Phi$. Let nodes $a_0, b_0, c_0, a_1, b_1, c_1, \ldots, a_r, b_r, c_r$ be given as in Definition 3, which implies the subprogram $\Phi_{\sigma,c_i}$ 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. *$\det(N[c_i, j])$ has $z$-degree at most $s \max(deg_z(q), d)$.*
3. *For all $j \in \{1, \ldots, r\}$, $(v)_{a_j} = q^{r-j} \cdot (-1)^{j+1} C_r(X,z)$.*

*Proof.* Observe that $\det(N) = \det(M(\Phi)_{|y=q}) = f_{|y=q} \equiv 0$. Hence the first property follows from Proposition 4. 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 $\det(N[c_r, a_j]) = \sum_C sgn(C)w(C)$, 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:

- From the source $\sigma$ there is a path to $c_r$.
- Then the edge $(c_r, a_j)$ is taken.
- Then $r - j$ edges with label $q$ are taken.
- Finally, the loop back from $a_r = \tau$ to the source $\sigma$ is taken.

- 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 $\sigma$ 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 $sgn(C) = (-1)^{j+1}$. Hence $\det(N[c_r, a_j]) = \sum_C sgn(C)w(C) = (-1)^{j+1}\sum_C w(C)$. The expression $\sum_C w(C)$ equals the sum of weights of all paths from $\sigma$ to $\tau$ 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 $\sigma$ to $c_r$) times $q^{r-j}$ (weights of path "$(c_r, a_j)$, followed by going from $a_j$ to $\tau$"). □

# 5 Main Lemma and its Proof

**Lemma 10.** *Let $f \in \mathbb{F}[X, y]$ be a nice polynomial of degree $r > 1$ computed by a standard form ABP $\Phi$ of size $s$. Suppose $f$ 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 in general position. Then any $p_i$ can be computed by an ABP of size $O((r^3d^7s^{12+\gamma_1} + r^{6+\gamma_2}d^5s^7)^{\gamma_3})$, where $d = \max_{i\in[r]} deg(p_i)$ and $\gamma_1, \gamma_2$ and $\gamma_3$ are the absolute constants introduced in Section 2.*

*Proof.* Our goal is to construct a small ABP for any desired root $p_i$. Wlog. we will show the method for obtaining $p_1$. The other roots follow by making a suitable variable renaming.

## 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 $f_{|y=0} \not\equiv 0$. Apply Lemma 8 to obtain matrices $A$ and $B$ of order $s$ such that $M(\Phi) = A - yB$. We have that

$$f = \det(A - yB).$$

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. Since $p_1, p_2, \dots, p_r$ are in general position, $z^d$ appears with a different coefficient in each $q_i$. Our aim is to exploit this fact in order to differentiate between the different $q_i$s.

Let $\alpha_i = coef(q_i, z^d)$. In other words, $\alpha_i = p_i(0)$. 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

$$f_\alpha = z^{d(s-r)} \cdot (q_1 - \alpha z^d - y)(q_2 - \alpha z^d - y)\dots(q_r - \alpha z^d - y).$$

Let $R = z^d A(\frac{x_1}{z}, \frac{x_2}{z}, \dots, \frac{x_n}{z})$, and $Q = B$.

**Lemma 11.** *The following three statements are true:*

1. $R - yQ = \overline{M(\Phi)}$.
2. $\forall\alpha \in \mathbb{F}$, $f_\alpha = \det(R - \alpha z^d Q - yQ)$. *In particular* $f_0 = \det(R - yQ)$.
3. $R$ *is nonsingular.*

*Proof.* The first property is clear. For the second property, note that Proposition 3 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. □

Let $\ell$ be the number of layers of $\Phi$. Note that $s > \ell > r$. Define for any $\alpha \in \mathbb{F}$, $\overline{f}_\alpha = f_\alpha / z^{d(s-\ell)}$. Then

$$\overline{f}_\alpha = z^{d(\ell-r)} \cdot (q_1 - \alpha z^d - y)(q_2 - \alpha z^d - y) \dots (q_r - \alpha z^d - y),$$

Let $\overline{\Phi}$ be the $d$-homogenization of $\Phi$. Note that $\overline{\Phi}$ is in generalized standard from. Observe, that $Q$ does not have nonzero entries on its diagonal or on the 'loopback' entry on row $\tau$, column $\sigma$, since in the standard form nodes labeled with the variable $y$ do not appear on self-loops or the loopback edge from $\tau$ to $\sigma$. By Proposition 2, we can write

$$M(\overline{\Phi}) = P - yQ,$$

where $P$ is obtained from $R$ by setting all nonzero diagonal entries and the 'loopback' entry $(\tau, \sigma)$ 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 2, 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. □

**Remark 1.** *Observe that by definition $\overline{\Phi}$ computes $\det(M(\overline{\Phi}))$, and that this equals $\det(P - yQ) = \overline{f}_0$, due to the above corollary.*

**Lemma 12.** *Let nodes $a_0, b_0, c_0, a_1, b_1, c_1, \dots, a_r, b_r, c_r$ in $\overline{\Phi}$ be given as in Definition 3. Let $C_i(X, z)$ be the polynomial computed by the subprogram $\overline{\Phi}_{\sigma, c_i}$. Clearly, we have that $\overline{f}_0 = \sum_{i=0}^{r} C_i(X, z) y^i$. 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 $N_{q_i} = M(\overline{\Phi})_{|y=q_i}$. In other words, $N_{q_i} = P - q_i Q$. Then the following four statements are true:*

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 $n - r$ standard basis vectors $e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}$ such that if we define the matrix $V$ by letting*

   $$V = [v_1, v_2, \dots, v_r, e_{i_1}, e_{i_2}, \dots, e_{i_{s-r}}],$$

   *then*

   $$\det(V) = (\pm 1) \cdot z^{d(\ell-r)r} \cdot \text{Vandet}(q_1, q_2, \dots, q_r).$$

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 $\gamma_2$ is the absolute constant introduced in Section 2.*

*Proof.* The first and second item immediately follow from Lemma 9, Items 1 and 2. Note that $C_r(X,z) = coef(y^r, \overline{f}_0) = (-1)^r z^{d(\ell-r)}$. By Lemma 9, Item 3, up to reordering of rows and multiplying rows with the field element $-1$, the matrix $V' = [v_1, v_2, \ldots, v_r]$ consisting of the $r$ column vectors $v_1, v_2, \ldots, v_r$ contains the following $r \times r$ Vandermonde matrix as a submatrix on rows in the set $J = \{a_j : j \in [r]\}$:

$$z^{d(\ell-r)} \cdot \begin{pmatrix} q_1^{r-1} & q_2^{r-1} & \ldots & q_r^{r-1} \\ \vdots & & & \\ q_1 & q_2 & \ldots & q_r \\ 1 & 1 & \ldots & 1 \end{pmatrix}$$

Choosing $e_{i_1}, e_{i_2}, \ldots, e_{i_{n-r}}$ to be an independent set of vectors that is zero on rows indexed by $J$ gives the third property. The fourth property will be proved in the next subsection. □

**Remark 2.** *As a note on the side, we observe that the vector $v_i$ is an eigenvector of $P^{-1}Q$ corresponding to eigenvalue $1/q_i$, when working over the field $\mathbb{G}$. Namely,*

$$\begin{aligned} Nv_i = 0 &\Leftrightarrow (P - q_i \cdot Q)v_i = 0 \\ &\Leftrightarrow (P^{-1}) \cdot (P - q_i \cdot Q)v_i = 0 \\ &\Leftrightarrow (I - q_i \cdot P^{-1}Q)v_i = 0 \\ &\Leftrightarrow P^{-1}Qv_i = 1/q_i \cdot v_i. \end{aligned}$$

## 5.2 A Small ABP for Computing $\mathrm{Vandet}(q_1, q_2, \ldots, q_r)^2$

This subsection is dedicated to proving Item 4 of Lemma 12. Define the polynomial

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

We use the fact[8] that

$$\det \begin{pmatrix} T_0 & T_1 & \ldots & T_{r-1} \\ T_1 & T_2 & \ldots & T_r \\ \vdots & & & \\ T_{r-1} & T_r & \ldots & T_{2r-2} \end{pmatrix} = \mathrm{Vandet}(x_1, x_2, \ldots, x_r)^2. \tag{1}$$

The strategy is to express each $T_i$ as a 'small' formula of $S_r^j(x_1, x_2, \ldots, x_r)$, where $S_r^j$ is the elementary symmetric polynomial in $r$ variables of degree $j$, i.e.

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

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 above[9], regardless of not having small ABPs for the $q_i$s, we readily have small ABPs for any $S_r^j(q_1, q_2, \ldots, q_r)$. Namely, consider the following remark and subsequent derivation:

**Remark 3.** *For every $j$, $S_r^j(p_1, p_2, \ldots, p_r)$ equals the coefficient of $y^{r-j}$ of $f$ modulo a factor of $\pm 1$. Hence an ABP $\Phi_j$ computing $S_r^j(p_1, p_2, \ldots, p_r)$ of size at most $s$ is easily obtained from the standard form ABP $\Phi$.*

[8] This follows by multiplying the Vandermonde matrix with nodes $x_1, x_2, \ldots, x_r$ by it transpose.

[9] In [13] the converse is achieved to get small depths formulas for $S_r^j$.

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

$$\begin{aligned}
S_r^j(q_1,q_2,\ldots,q_r) &= S_r^j(z^d p_1(\frac{x_1}{z},\frac{x_2}{z},\ldots,\frac{x_n}{z}),\ldots,z^d p_r(\frac{x_1}{z},\frac{x_2}{z},\ldots,\frac{x_n}{z}))\\
&= z^{dj} S_r^j(p_1(\frac{x_1}{z},\frac{x_2}{z},\ldots,\frac{x_n}{z}),\ldots,p_r(\frac{x_1}{z},\frac{x_2}{z},\ldots,\frac{x_n}{z}))\\
&= z^{dj}(\pm 1)\cdot D_j(\frac{x_1}{z},\frac{x_2}{z},\ldots,\frac{x_n}{z})
\end{aligned}$$

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

### 5.2.1 A Formal Power Series Identity Related to the Newton Identities

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

**Lemma 13.** *Provided the characteristic of $\mathbb{F}$ is zero, we have the following identity in the ring of formal power series $\mathbb{F}[[X]]$: $\sum_{\ell\geq 1}\frac{1}{\ell}(\sum_{j=1}^r(-1)^j S_r^j(x_1,x_2,\ldots,x_r)w^j)^\ell = \sum_{n\geq 1}\frac{-w^n}{n}T_n(x_1,x_2,\ldots,x_r)$.*

*Proof.* We recall the definitions of the formal power series (FPS) $exp(w)$ and $\log(1-w)$. These are given by $\exp(w)=\sum_{n\geq 0}\frac{w^n}{n!}$. 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

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

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

$$\sum_{j=1}^r(-1)^j S_n^j(x_1,x_2,\ldots,x_r)w^j = \exp(-\sum_{n\geq 1}T_n(x_1,x_2,\ldots,x_r)w^n/n) - 1.$$

Now we use that for $g(w) := -\sum_{n\geq 1}T_n(x_1,x_2,\ldots,x_r)w^n/n$, 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

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

□

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

1. *There exists an ABP $\Gamma(u_1, u_2, \ldots, u_i, w)$ with $O(i^3)$ many edges computing $E$.*
2. *There exists an ABP $\Gamma'(u_1, u_2, \ldots, u_i)$ with $O(i^4)$ many edges computing the coefficient of $w^i$ in $E$.*
3. *Say $\Gamma'$ computes the polynomial $E'$. Then $E'(S_r^1(x), S_r^2(x), \ldots, S_r^i(x)) = -T_i(x_1, x_2, \ldots, x_r)/i$.*

*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 13. □

### 5.2.2 Putting It Together

By Proposition 5, $E'(S_r^1(q_1, \ldots, q_r), S_r^2(q_1, \ldots, q_r), \ldots, S_r^i(q_1, \ldots, q_r)) = -T_i(q_1, \ldots, q_r)/i$. By Remark 3 and comments thereafter, we conclude that for any $i \in [r]$, we have an ABP computing $T_i(q_1, \ldots, q_r)$ 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 (1) we obtain an ABP for computing $\text{Vandet}(q_1, q_2, \ldots, q_r)^2$ of size $O(r^{5+\gamma_2} ds)$.

## 5.3 Selecting a Good Starting Vector $u$

As hinted on before, our strategy is to use $P$ and $Q$ to find the root $q_1$ of $f_0$. We will use an iterative method, where we repeatedly apply the matrices $Adj(P)$ and $Q$ to some chosen starting vector $u$. As will become clear later, we must pick this $u$ to be of low degree in order for our method to converge within reasonable number of iterations. We take care of this next. Let

$$M_0 = Adj(P)Q.$$

and let $s \times s$ matrix $V = [v_1, v_2, \ldots, v_r, e_{i_1}, e_{i_2}, \ldots, e_{i_{s-r}}]$ be given by Lemma 12. The set $\{M_0 v : v \in \mathbb{G}^s\}$ we denote by $Range(M_0)$.

**Proposition 6.** *Working over the field $\mathbb{G}$, we have the following properties:*

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

*Proof.* By Corollary 1, the polynomials $q_1, q_2, \ldots, q_r$ are precisely all the solutions for $y$ of the equation $\det(P - yQ) \equiv 0$. For a polynomial $q$, we have that

$$\det(P - q \cdot Q) \equiv 0 \quad \Leftrightarrow \quad \exists v \in \mathbb{G}^s \neq \vec{0}, \text{ such that } Pv = q \cdot Qv.$$

Lemma 12 shows that for every $i \in [r]$,

$$(P - q_i Q)v_i = 0.$$

Due to Item 2 of Corollary 1, $Adj(P)$ is nonsingular. Hence this is equivalent to

$$(\det(P)I - q_i M_0)v_i. = 0 \tag{2}$$

Since $q_i \not\equiv 0$, 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 12 gives that $v_1, v_2, \ldots, v_r$ are independent vectors. Due to Item 2, Lemma 8, $rank(Q) = r$. Since $Adj(P)$ is nonsingular, we have that $rank(M_0) = rank(Q) = r$. Hence it must be that $v_1, v_2, \ldots v_r$ form a basis of $Range(M_0)$. □

**Remark 4.** *The following observation is not used in the sequel. The characteristic polynomial of $M_0$ is given by*

$$p(\lambda) = \det(\lambda I - M_0).$$

*This polynomial is an element of $\mathbb{G}[\lambda]$. The above show that each $\lambda - \frac{\det(P)}{q_i}$ divides $p(\lambda)$. We have that $dim(null(M_0)) = s - r$, since $rank(M_0) = r$. By linear algebra, the geometric multiplicity of an eigenvalue always is less than or equal to the algebraic multiplicity. Hence $\lambda^{s-r}$ divides $p(\lambda)$. Since $deg_\lambda(p) = s$ and all of $0, q_1, q_2, \ldots, q_r$ are distinct this implies that*

$$p(\lambda) = \lambda^{s-r}(\lambda - \frac{\det(P)}{q_1})(\lambda - \frac{\det(P)}{q_2})\ldots(\lambda - \frac{\det(P)}{q_r}).$$

**Lemma 14.** *There exists $i \in [s]$ such that we can write*

$$\det(V)^2 M_0 e_i = a_1 v_1 + a_2 v_2 + \ldots + a_r v_r,$$

*where*

1. $\forall i, a_i \in \mathbb{F}[X, z]$.
2. $a_1 \neq 0$.
3. $\forall i$, $deg_z(a_i) \leq d^3 s^5$.

*Proof.* By Item 2 of Proposition 6, $v_1, v_2, \ldots, v_r$ forms a basis of $range(M_0)$. Hence for every $e_i$, we can write

$$\det(V)^2 M_0 e_i = a_{1,i} v_1 + a_{2,i} v_2 + \ldots + a_{r,i} v_r,$$

for certain $a_{1,i}, a_{2,i}, \ldots, a_{r,i} \in \mathbb{G}$. Suppose that for every $i \in [s]$, $a_{1,i} = 0$. This means that $range(M_0) \subseteq span(v_2, \ldots, v_r)$, i.e. $rank(M_0) \leq r - 1$. This is a contradiction, as we observed before that $rank(M_0) = r$. Now let $i$ be such that $a_{1,i} \neq 0$. The coefficients $a_{1,i}, a_{2,i}, \ldots, a_{r,i}$ can be obtained as the first $r$ components of the vector

$$V^{-1} \det(V)^2 M_0 e_i = \det(V) \cdot Adj(V) M_0 e_i.$$

Note that this implies all $a_i$ are in $\mathbb{F}[X, z]$, as all of $V, Adj(V)$ and $M_0$ only have polynomial entries. The $z$-degrees of entries in $Adj(V)$, 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 12. Furthermore, $d$ bounds the $z$-degrees of entries of $P$. So $Adj(P)$ 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. □

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

*Proof.* By Lemma 12, 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 $M_0 = Adj(P)Q$. By Item 2, Lemma 8, $Q$ is a projection. Therefore $M_0 e_i$ equals some column of $Adj(P)$. Each entry of $Adj(P)$ 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. □

## 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

$$u = a_1 v_1 + a_2 v_2 + \ldots + a_r v_r,$$

for certain $a_i \in \mathbb{F}[X, z]$, as given by Lemma 14. Together with Lemma 12, 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

$$M_\alpha = Adj(P_\alpha)Q.$$

Note this definition coincides with previously defined $M_0$. We have the following proposition:

**Proposition 7.** *For all $\alpha \in \mathbb{F}$, $i \in [r]$, $P_\alpha v_i = (q_i - \alpha z^d) \cdot Q v_i$.*

*Proof.* By Lemma 12, Item 1, it holds that $P v_i = q_i Q v_i$, for any $i \in [r]$. Hence

$$\begin{aligned} P_\alpha v_i &= P v_i - \alpha z^d Q v_i \\ &= q_i Q v_i - \alpha z^d Q v_i \\ &= (q_i - \alpha z^d) \cdot Q v_i. \end{aligned}$$

□

First we consider what happens when we apply $M_\alpha$ to one of the eigenvectors $v_1, v_2, \ldots, v_r$.

**Proposition 8.** *For all $i \in [r]$, $e \geq 1$, $M_\alpha^e v_i = \left(\frac{\det(P_\alpha)}{(q_i - \alpha z^d)}\right)^e \cdot v_i$. Furthermore, the entries of $M_\alpha v_i$ lie in $\mathbb{F}[X, z]$.*

*Proof.* Since $P_\alpha v_i = (q_i - \alpha z^d) \cdot Q v_i$, we have that

$$\begin{aligned} \det(P_\alpha) v_i &= Adj(P_\alpha) P_\alpha v_i \\ &= Adj(P_\alpha)(q_i - \alpha z^d) \cdot Q v_i \\ &= (q_i - \alpha z^d) M_\alpha v_i. \end{aligned}$$

Hence, since $q_i - \alpha z^d \not\equiv 0$, 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 $M_\alpha$ both lie in $\mathbb{F}[X, z]$. □

More generally, we have the following statement:

**Proposition 9.** *Given $\alpha_2, \alpha_3, \ldots, \alpha_r \in \mathbb{F}$, for all $i \in [r]$, $e \geq 1$ we have that*

$$M_{\alpha_r}^e \ldots M_{\alpha_3}^e M_{\alpha_2}^e v_i = \prod_{j=2}^{r} \left(\frac{\det(P_{\alpha_j})}{(q_i - \alpha_j z^d)}\right)^e \cdot v_i,$$

*and the entries of $M_{\alpha_r}^e \ldots M_{\alpha_3}^e M_{\alpha_2}^e v_i$ lie in $\mathbb{F}[X, z]$. Consequently, for any $\ell \in [s]$, we have that*

- *if* $(v_i)_\ell \not\equiv 0$*, then the* $z$*-degree of the* $\ell$*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 8. 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

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

Hence

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

□

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 $M_\alpha$, we have that

$$\begin{aligned} M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e u &= \sum_{i \in [r]} a_i M_{\alpha_r}^e \dots M_{\alpha_3}^e M_{\alpha_2}^e v_i \\ &= \sum_{i \in [r]} a_i \prod_{j=2}^{r} \left( \frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i, \end{aligned}$$

where the last equation follows from Proposition 9.

Let

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

By Proposition 9, 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. $\eta$ 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 14 established that $a_1 \not\equiv 0$. We let $R$ equal the maximum $z$-degree of any nonzero component of the $i = 1$ term in (3), i.e. 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 $\alpha_1$, 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 9 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 $a_i \not\equiv 0$, let $T_i$ be he minimum $z$-degree of any nonzero component of the $i$th term of (3), i.e. let

$$T_i = mindeg_z \left( a_i \prod_{j=2}^{r} \left( \frac{\det(P_{\alpha_j})}{q_i - \alpha_j z^d} \right)^e v_i \right)$$

Note that for $q_i - \alpha_j z^d$, if $j = i$, we do not have the maximum degree monomial $z^d$ appearing. Proposition 9 therefore gives us that

$$\begin{aligned} T_i &\geq \sum_{j=2}^{r} e \cdot deg_z(\det(P_{\alpha_j})) - \sum_{j=2}^{r} e \cdot deg_z(q_i - \alpha_j z^d) \\ &\geq \sum_{j=2}^{r} e \cdot deg_z(\det(P_{\alpha_j})) - e(r-2)d - e(d-1) \\ &\geq \sum_{j=2}^{r} e \cdot deg_z(\det(P_{\alpha_j})) - e(r-1)d + e. \end{aligned}$$

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 $a_i \not\equiv 0$. Recall that we observed before that $deg_z(a_1) + deg_z(v_1) \leq d^3 s^5 + ds$. This provides us with a bound on how large $e$ needs to be set. 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

$$[g]_{\leq_z \kappa} \quad = \quad a_1 \prod_{j=2}^{r} \left( \frac{\det(P_{\alpha_j})}{q_1 - \alpha_j z^d} \right)^e v_1. \tag{4}$$

## 5.5 Constructing the ABP for the Eigenvalue $q_1$

**Lemma 16.** *The vector $g = M^e_{\alpha_r} \dots M^e_{\alpha_3} M^e_{\alpha_2} u$ 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 15 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 $M_\alpha$, for $\alpha \in \mathbb{F}$. Each such matrix multiplication can be achieved by adding $O(rs^{1+\gamma_1})$ nodes to the ABP. Namely, $M_\alpha = Adj(P_\alpha)Q = Adj(P - \alpha z^d Q)Q$. By Item 2, Lemma 8, Q is a projection, i.e. $M_\alpha$ consists of $r$ columns selected from $Adj(P - \alpha z^d Q)$. If we would allow arbitrary polynomials on the wires of ABPs, this means that multiplication by $M_\alpha$ 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 $Adj(P - \alpha z^d Q)$. Within the generalized ABP model we can achieve the same, by expanding each such edge into a subprogram computing the appropriate entry of $Adj(P - \alpha z^d Q)$. By Theorem 3, each entry of $Adj(P - \alpha z^d Q)$ 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 $M_\alpha$.

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 $r < s$. □

Now we can use Lemma 4, to discard the homogeneous components of $g$ of degree larger than $\kappa$ to obtain the following corollary:

**Corollary 2.** $[g]_{\leq_z \kappa}$ *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 $\tilde{v_1} = [g]_{\leq_z \kappa}$, i.e.

$$\tilde{v_1} := a_1 \prod_{j=2}^{r} \left( \frac{\det(P_{\alpha_j})}{q_1 - \alpha_j z^d} \right)^e v_1.$$

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 $\ell$ is such that $(v_1)_\ell$ is a nonzero component (which must exist), we get that

$$\frac{(\tilde{v_1})_\ell \cdot \det(P)}{(M_0 \tilde{v_1})_\ell} = q_1. \tag{5}$$

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 of the resulting generalized ABP 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 $M_0$, and then simply selecting the subprogram computing the $\ell$th component. 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 for the above two ABPs. We conclude $q_1$ 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 shows how to extract the root $p_1$, and completes the proof of Main Lemma 10. □

# 6 Main Theorem

We restate the main theorem from the introduction:

**Theorem 5** (Theorem 1 Restated)**.** *Let $\mathbb{F}$ be a field of characteristic zero. Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n, y]$ be a nonzero polynomial that can be computed by an ABP of size $s$. Suppose $f$ factors as*

$$f = p_0(p_1 - y)(p_2 - y) \ldots (p_r - y),$$

*where $\{p_0, p_1, p_2, \ldots, p_r\} \subset \mathbb{F}[x_1, x_2, \ldots, x_n]$ and $p_1, p_2, \ldots, p_r$ are pairwise distinct. Then every $p_i$ has an ABP of size at most polynomial in $d, r$ and $s$, where $d = \max_{i \in [r], p_i \not\equiv 0} deg(p_i)$.*

*Proof.* First we convert the ABP for $f$ to standard form using Lemma 7. This blows up the size to $O(r^2 s)$. Next we apply Lemma 10. The composition of these two operations yields that for the function $\beta(r, d, s)$ from Lemma 6 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 $\gamma_1, \gamma_2$ and $\gamma_3$ are the absolute constants introduced in Section 2. To be concrete, by Theorem 3 and Lemma 2, we can take $\gamma_1 = 3, \gamma_2 = 5$ and $\gamma_3 = 12$. Hence we have $\beta(r, d, s) = O((r^{33} d^7 s^{15} + r^{25} d^5 s^7)^{12})$. Hence we can write $\beta(r, d, s) = O(r^{396} d^{84} s^{180})$. Hence by Lemma 6, every $p_i$ can be computed by an ABP of size $\beta(r, d, (rs)^{\gamma_3}) = O(r^{2556} d^{84} s^{2160})$. □

# 7 Roots of General Arithmetic Circuits and Newton's Method

For a univariate polynomial $f(y) \in \mathbb{R}[y]$ with $f(p) = 0$ for $p \in \mathbb{R}$, Newton's method is to start with some initial guess $y_0$ for $p$, and to compute successive (hopefully) better approximations $y_1, y_2, \ldots$ according to the rule

$$y_{k+1} = y_k - \frac{f(y_k)}{f'(y_k)},$$

where $f'$ is the derivative of $f$. For arithmetic circuits we have the following analogue, where we compute successively better approximations $p_{\leq k}$, $p_{\leq k+1}, \ldots$ to a root $p \in \mathbb{F}[X]$ of $f \in \mathbb{F}[X, y]$.

**Lemma 17.** *Let $f \in \mathbb{F}[X, y]$ be such that $\deg_y(f) = r$. Write $f = \sum_{i=0}^{r} C_i(x)y^i$, and let $f'(x, y) := \frac{\partial f}{\partial y} = \sum_{i=1}^{r} iC_i(x)y^{i-1}$. Let $p \in \mathbb{F}[X]$ be such that $f(x, p) = 0$ and assume that $\xi_0 := f'(0, p(0)) \neq 0$. Then for any integer $k \geq 1$ it holds that*

$$p_{\leq k+1} = p_{\leq k} - \frac{1}{\xi_0} \cdot f(x, p_{\leq k})_{=k+1}.$$

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

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

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. □

**Corollary 3** (Theorem 2 Restated)**.** *Let $\mathbb{F}$ be a field of characteristic zero. Let $f \in \mathbb{F}[X, y]$ 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. $f(x_1, x_2, \ldots, x_n, p) \equiv 0$ 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 $f$.*

*Proof.* In case $f'(0, p(0)) \neq 0$, we can construct an arithmetic circuit for $p$ by repeatedly applying Lemma 17. 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 $p_0, p_1, \ldots, p_{=k}$ computed at gates somewhere already, we use a copy of a circuit $\Phi$ 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 $v_0, v_1, \ldots, v_{k+1}$ be the gates in $\Phi$ corresponding to the output gate of the original circuit, i.e. $f_0, f_1, \ldots, f_{=k+1}$ are computed at these gates. We can bound the size of $\Phi$ 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 when homogeneous components are given separately. Note that having $p_0, p_1, \ldots, p_k$ computed separately at gates is exactly the right format for feeding $p_{\leq k}$ into $\Phi$ for the variable $y$. Namely, for any nodes $w_0, w_1, \ldots, w_{k+1}$ that correspond[10] to the splitting of an input node labeled by $y$, for any $i$, replace any edge $(w_i, u)$ by an edge from the gate computing $p_{=i}$ to $u$. 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 17 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

[10] These are just input nodes computing 0, except for $w_1$, which computes $y$.

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 $C_r(x) \not\equiv 0$. Let $f^i(x, y) = \frac{\partial^i f}{\partial^i y}$. Then $f^r(x, y) = r! \cdot C_r(x)$. Since the characteristic of $\mathbb{F}$ is zero, $r! \neq 0$, so $f^r(x, p) \not\equiv 0$. We have in this case that $f^0(x, p) \equiv 0$. Let $i$ be the smallest integer for which $f^i(x, p) \not\equiv 0$. Then $0 < i \leq r$, and $f^{i-1}(x, p(x)) \equiv 0$. Due to Lemma 1, $y - p$ is an irreducible factor of $f$.

**Claim 1.** *The number $i$ equals the multiplicity of the root $p$ in $f$.*

*Proof.* Let $m$ be the largest number for which we can write $f = (p - y)^m \cdot h$, for $h \in \mathbb{F}[X, y]$, i.e. $m$ is the multiplicity of the root $p$ in $f$. We have that $p - y$ does not divide $h$. Repeatedly computing partials yields:

$$\frac{\partial f}{\partial y} = -m(p - y)^{m-1} \cdot h + (p - y)^m \frac{\partial h}{\partial y}.$$

$$\frac{\partial^2 f}{\partial^2 y} = m(m - 1)(p - y)^{m-2} \cdot h - m(p - y)^{m-1} \frac{\partial h}{\partial y} - (p - y)^{m-1} \frac{\partial h}{\partial y}. + (p - y)^m \frac{\partial^2 h}{\partial^2 y}.$$

Note in the latter equation each term contains a factor $(p - y)^d$, where $d \geq m - 2$. More generally, for $k < m$, one obtains an expression for $\frac{\partial^k f}{\partial^k y}$ as a sum a terms, where each term contains the factor $(p - y)^d$, where $d \geq m - k$. Hence for $k < m$, $p - y$ divides $\frac{\partial^k f}{\partial^k y}$. This implies that for $k < m$, $f^k(x, p) \equiv 0$. Similarly, we get that

$$f^m(x, p) = \left( \frac{\partial^m f}{\partial^m y} \right)_{|y:=p} = ((-1)^m m! \cdot h)_{|y:=p} .$$

We know that $h_{|y:=p} \not\equiv 0$, since, if not, Lemma 1 would yield that $p - y$ divides $h$, which is a contradiction. □

We have that there exists $x_0 \in \mathbb{F}$ such that $f^i(x_0, p(x_0)) \neq 0$, e.g. by Lemma 5. 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 $O(\mathcal{M}(m)\mathcal{M}(deg(p)) \cdot deg(p) \cdot s)$. □

Lemma 1 implies that for the situation of the above corollary, $\max(m, deg(p)) \leq deg(f)$. From this we draw the following conclusion:

**Corollary 4.** *The class* VP *is closed under taking roots.*

# References

[1] S. Toda. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Trans. Inf. Syst.*, E75-D:116–124, 1992.

[2] Erich Kaltofen. Factorization of polynomials given by straight-line programs. In *Randomness and Computation*, pages 375–412. JAI Press, 1989.

[3] P. Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004.

[4] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity testing means proving circuit lower bounds. *Computational Complexity*, 13(1–2):1–44, 2004.

[5] 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.

[6] 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.

[7] D.G. Cantor and E. Kaltofen. On fast multiplication of polynomials over arbitrary algebras. *Acta Informatica*, 28(7):693–701, 1991.

[8] 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.

[9] M. Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chicago Journal of Theoretical Computer Science*, 1997(Article 5), 1997.

[10] N. Alon. Combinatorial nullstellensatz. *Combinatorics, Probability and Computing*, 8(1–2):7–29, 1999.

[11] 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.

[12] L. Valiant. Completeness classes in algebra. In *Proc. 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261, 1979.

[13] A. Shpilka and A. Wigderson. Depth-3 arithmetic formulae over fields of characteristic zero. *Journal of Computational Complexity*, 10(1):1–27, 2001.