

Multiplicity Problems on Algebraic Series and Context-Free Grammars

Nikhil Balaji*, Lorenzo Clemente†, Klara Nosan‡, Mahsa Shirmohammadi‡ and James Worrell§

*IIT Delhi, India

†University of Warsaw, Poland

‡Université Paris Cité, CNRS, IRIF, France

§Department of Computer Science, University of Oxford, UK

Abstract—In this paper we obtain complexity bounds for computational problems on algebraic power series over several commuting variables. The power series are specified by systems of polynomial equations: a formalism closely related to weighted context-free grammars. We focus on three problems—decide whether a given algebraic series is identically zero, determine whether all but finitely many coefficients are zero, and compute the coefficient of a specific monomial. We relate these questions to well-known computational problems on arithmetic circuits and thereby show that all three problems lie in the counting hierarchy. Our main result improves the best known complexity bound on deciding zeroness of an algebraic series. This problem is known to lie in PSPACE by reduction to the decision problem for the existential fragment of the theory of real closed fields. Here we show that the problem lies in the counting hierarchy by reduction to the problem of computing the degree of a polynomial given by an arithmetic circuit. As a corollary we obtain new complexity bounds on multiplicity equivalence of context-free grammars restricted to a bounded language, language inclusion of a non-deterministic finite automaton in an unambiguous context-free grammar, and language inclusion of a non-deterministic context-free grammar in an unambiguous finite automaton.

I. INTRODUCTION

The subject of this paper is algebraic power series—formal series that satisfy a polynomial equation over the field of rational functions. For example, consider the generating function $C(x) := \sum_{n=0}^{\infty} C_n x^n$ of the sequence $(C_n)_{n=0}^{\infty}$ of Catalan numbers. Recall that C_n is the number of Dyck words of length $2n$. The series $C(x)$ satisfies the polynomial equation $1 - C(x) + xC(x)^2 = 0$ and hence is algebraic. Algebraic power series generalise rational series (which are the generating functions of linear recurrence sequences) and are a subclass of D-finite power series (which are the generating functions of holonomic sequences). To illustrate the latter inclusion, note that the Catalan numbers satisfy the holonomic recurrence $(n+2)C_{n+1} = 2(2n+1)C_n$.

Algebraic power series have been an object of study in formal language theory ever since the seminal work of Chomsky and Schützenberger (see [1, Chapter IV] and [2, Chapters II

and II] for the principal results and bibliographic references). In this framework one specifies algebraic series via so-called *proper systems of polynomial equations*, which can alternatively be seen as weighted context-free grammars or as a generalisation of arithmetic circuits that allows cycles. Such equation systems can be defined both over commuting and non-commuting variables. In this work we focus on equation systems over several commuting variables with weights in \mathbb{Z} .

Two fundamental computational problems associated with an algebraic series are to compute its coefficients and to determine whether or not the series is identically zero. These are generalisations of two well-studied problems on circuits, namely the problem CoeffSLP of determining a given coefficient of the polynomial represented by an arithmetic circuit, and the problem EqSLP of deciding zeroness of such a polynomial. (The acronym SLP here stands for *straight line program*, which we treat as synonymous with the term arithmetic circuit. The problem EqSLP is more commonly called *arithmetic circuit identity testing* or *polynomial identity testing*.)

We extend the problem CoeffSLP from circuits to algebraic series, obtaining the problem CoeffAlg. The input to the latter is a system of polynomial equations whose unique solution is a multivariate power series $\sum_{\mathbf{v} \in \mathbb{N}^k} a_{\mathbf{v}} X^{\mathbf{v}}$ with integer coefficients, a vector $\mathbf{v} \in \mathbb{N}^k$, and a prime number p (where \mathbf{v} , p , and all coefficients of the system of polynomial equations are given in binary). The problem asks to determine $a_{\mathbf{v}} \bmod p$. The reason to introduce the modulus p is because (as already in the circuit case) the bit-length of $a_{\mathbf{v}}$ is potentially exponential in the bit-length of \mathbf{v} . *and exp in ~~xx~~ equations*

Our approach to CoeffAlg involves a multivariate version of Hensel’s Lemma for computing zeros of systems of polynomials in the ring of formal power series. The advantage of this approach is that, thanks to the quadratic convergence of the root approximation in Hensel’s Lemma (meaning that the precision of the approximation doubles with each iteration), one can compute the n -th coefficient of a power series using a number of arithmetic operations that is polynomial in the bit length of n . We exploit this fact to reduce CoeffAlg to CoeffSLP. The latter problem is known to be $\#P$ -hard (see for example [3]) and from the proof of [4, Theorem 4.1] it can be deduced to be in $\mathbf{FP}^{\#P}$. We thus have:

Theorem 1. CoeffAlg is equivalent under polynomial time

Nikhil Balaji and Mahsa Shirmohammadi are supported by International Emerging Actions grant (IEA’22). Mahsa Shirmohammadi is supported by ANR grant VeSyAM (ANR-22-CE48-0005). James Worrell is supported by EPSRC fellowship EP/X033813/1. Lorenzo Clemente is partially supported by the Polish NCN grant 2017/26/D/ST6/00201 and by the European Research Council (ERC) project INFSYS (“Challenging Problems in Infinite-State Systems”, grant agreement No. 950398).

reductions to CoeffSLP and hence is $\#P$ -hard and in $FP^{\#P}$.

To the best of our knowledge, the theoretical complexity of EqAlg has not been studied before.

Our second main result concerns the complexity of the problem EqAlg: *given a proper system of polynomial equations, determine whether all coefficients of its power series solution are zero*. This problem was shown in [1, Chapter IV, Theorem 5.1] to be polynomial-time reducible to the decision problem for the theory of real-closed fields and later in [5, Lemma 1] to be reducible to the decision problem for the existential fragment of this theory, which we denote $\exists\mathbb{R}$. Here we improve the complexity upper bound on EqAlg by showing that it is polynomial-time reducible to the problem DegSLP: *given a polynomial f represented by an algebraic circuit and an integer d , decide whether $\deg f \leq d$* . The substance of this improvement is twofold: first, DegSLP is polynomial-time reducible to $\exists\mathbb{R}$ ¹; second, DegSLP is known to lie in the counting hierarchy, whereas the best upper bound for $\exists\mathbb{R}$ is $PSPACE$. It is also worth noting that DegSLP is not known to be NP -hard, whereas $\exists\mathbb{R}$ is trivially so.

Theorem 2. EqAlg is polynomial-time reducible to DegSLP and thereby lies in the counting class $coRP^{PP}$.

The proof of Theorem 2 combines Hensel's Lemma (as in Theorem 1) with an upper bound on the degree of an annihilating polynomial of an algebraic series specified by a system S of polynomial equations. The latter bound is singly exponential in the size of the system S , and is obtained using results about quantifier elimination over the theory of real-closed fields. The paper [6, Section 5] states a doubly exponential degree bound on the annihilating polynomial, and leaves open the existence of a singly exponential bound.

A third natural computational problem on algebraic series is FinAlg: *given a proper system of polynomial equations, determine whether denoted series has finite support*. To the best of our knowledge, the theoretical complexity of FinAlg has not been studied before. Concerning this problem we prove:

Theorem 3. The complement of FinAlg is non-deterministic polynomial-time reducible to CoeffSLP, and thereby FinAlg lies in $coNP^{PP}$.

The problems EqAlg and FinAlg have a natural characterisation in terms of context-free grammars. Associated with a context-free grammar over a k -element alphabet we have the *census generating function* (see [7])

$$f(x_1, \dots, x_k) := \sum_{\mathbf{v} \in \mathbb{N}^k} a_{\mathbf{v}} x_1^{v_1} \cdots x_k^{v_k},$$

where $a_{\mathbf{v}}$ is the number of leftmost derivations that produce a word with Parikh image \mathbf{v} . Then EqAlg is polynomial-time equivalent to the problem of whether two grammars have identical census generating functions, whereas FinAlg

is polynomial-time equivalent to the problem of whether two grammars have respective census generating functions that differ in only finitely many entries. Specialising EqAlg to the case of unary grammars we obtain the following corollary of Theorem 2.

Corollary 4. *Multiplicity equivalence of unary context-free grammars is in $coRP^{PP}$.*

Some consequences of the corollary above are worth mentioning. First, the universality problem for (not necessarily unary) unambiguous context-free grammars is in $coRP^{PP}$, since it reduces to multiplicity equivalence over a unary alphabet. This problem was previously known to be in $PSPACE$ ([5, Theorem 2] and [8, Theorem 10]). In turn, by the reductions in [8, Theorems 8 and 9] the following two problems are also in $coRP^{PP}$: deciding language inclusion of 1) a nondeterministic finite automaton in an unambiguous context-free grammar, and 2) a nondeterministic context-free grammar in an unambiguous finite automaton. The last two problems were known to be in $PSPACE$ [8].

Corollary 5, below, generalises Corollary 4 from the unary case to the more general case of letter-bounded context-free languages and context-free languages restricted to a given bounded language; we formally define these classes of languages in Sec. V. In both cases the proof goes via a deterministic polynomial-time reduction to EqAlg, and thus to DegSLP by Theorem 2.

Corollary 5. *The following two problems lie in $coRP^{PP}$:*

- 1) *Multiplicity equivalence of context-free grammars recognising letter-bounded languages;*
- 2) *Multiplicity equivalence of context-free grammars restricted to a given bounded language $w_1^* \cdots w_k^*$.*

A. Related Work

Hensel's Lemma is based on Newton iteration applied to the ring of multivariate power series. However, while the classical analysis of Newton's iteration provides convergence bounds with respect to the usual Euclidean metric of \mathbb{R}^k , Hensel's Lemma gives bounds in terms of an ultrametric that is more suitable to the context of power series. Thus our approach is related to the scheme of so-called Newtonian program analysis [9], which involves the use of Newton's method to solve polynomial equations in a variety of different semirings. The method has been applied to interprocedural dataflow analysis as well as to computing termination and reachability probabilities in quasi-birth-death processes [10], multi-type branching processes [11], and stochastic grammars [12], [13].

Aside from language theory, algebraic series are an essential tool in combinatorics, where they are used to derive growth estimates on various types of combinatorial objects. In this context, Flajolet and Soria [14, Theorem 1] have developed an explicit formula for the n -th coefficient of an algebraic power series. A disadvantage of this formula for obtaining complexity

¹The reduction follows from [3], where DegSLP is shown to be polynomial-time reducible to the problem PosSLP (decide whether a variable-free arithmetic circuit denotes a positive integer), which in turn belongs to $\exists\mathbb{R}$.

pp^P = PP

bounds for CoeffAlg is that it requires first computing an annihilating polynomial for the series, and when algebraic series are succinctly encoded as *systems* of polynomial equations annihilating polynomials require exponential degree in general. Note also that if one has to handle the holonomic recurrence satisfied by the sequence of coefficients of an algebraic series, one can compute the coefficients one-by-one. However, even setting aside the overhead of computing such a recurrence, it is not clear to us whether one can prove Theorem 1 via this route.

II. BACKGROUND

A. Complexity Theory

We briefly summarise some relevant notions from complexity theory (see [15, Chapter 7] for more details). The class **FP** is the function problem version of the decision problem class **P**. Let Σ be a finite alphabet. The class **#P** is the collection of functions $f : \Sigma^* \rightarrow \mathbb{N}$ for which there is a non-deterministic polynomial-time Turing machine M such that $f(x)$ is the number of accepting computations of M on input x . The class **PP** (probabilistic polynomial time) is a decision analog of **#P**. A language $L \subseteq \Sigma^*$ is in **PP** if there is a non-deterministic polynomial-time Turing machine M such that $x \in L$ if and only if on input M at least one half of the computations of M on input x are accepting. The *counting hierarchy* is the family of complexity classes inductively defined by $C_0 := \mathbf{P}$ and $C_{k+1} := \mathbf{PP}^{C_k}$. In particular, the complexity class **coRP^{PP}** is included in C_2 : the second level of the counting hierarchy. We have $\bigcup_k C_k \subseteq \mathbf{PSPACE}$.

B. Polynomials and Arithmetic Circuits

Let $X := (x_1, \dots, x_k)$ be a tuple of indeterminates. Given $\mathbf{v} = (v_1, \dots, v_k)$ in \mathbb{N}^k , we denote by $X^{\mathbf{v}}$ the monomial $x_1^{v_1} \cdots x_k^{v_k}$. The *total degree* of $X^{\mathbf{v}}$ is defined to be $|\mathbf{v}| := \sum_{i=1}^k v_i$. We denote by $\mathbb{Z}[X]$ the (commutative) ring of polynomials with integer coefficients and (commuting) variables in X . The total degree of a polynomial is the maximum total degree of its constituent monomials. The *size* of a polynomial is the number of bits required to represent it when its coefficients and degrees are written in binary.

An *arithmetic circuit* over variables X is a directed acyclic graph with input gates labelled with the constants 0, 1 or with the indeterminates x_i . Internal gates are labelled with one of the operations $+$, $-$, \times and there is a distinguished output gate. Each gate of such a circuit represents polynomial in $\mathbb{Z}[X]$ that is computed in an obvious bottom-up manner starting from the input gates.

The *size* of a circuit is the number of its gates; see Figure 1. The following proposition shows that circuits can be exponentially more succinct than polynomials.

Proposition 6. *Given $m \in \mathbb{N}$, there is a circuit of size $O(\log m)$ that represents the polynomial $\sum_{i=0}^m x^i$.*

Conversely, the total degree and the bit-length of the coefficients of a polynomial represented by a circuit is always at

most exponential in the size of the circuit. The following is well-known [16], [17]:

Proposition 7. *Let $m \in \mathbb{N}$. Given an $m \times m$ matrix whose entries are distinct indeterminates, there is an algorithm that runs in **poly**(m) time and produces an algebraic circuit that represents the determinant of the matrix.*

A *straight-line program* (SLP) is a sequence of instructions corresponding to the sequential evaluation of an arithmetic circuit. In this paper, we treat arithmetic circuits and SLPs as synonymous. The following computational problems for arithmetic circuits are well-studied, see [4], [18], [3]. Unless otherwise stated, all integers are represented in binary and all polynomials are multivariate.

- **EqSLP**: Given an arithmetic circuit computing a polynomial f , decide whether f is the zero polynomial.²
- **DegSLP**: Given a positive integer d and an arithmetic circuit computing a polynomial f , decide whether the total degree of f is at most d .
- **CoeffSLP**: Given a multi-index \mathbf{v} and prime p (both encoded in binary), and an arithmetic circuit computing a polynomial $f(X)$, compute the residue modulo p of the coefficient of the monomial $X^{\mathbf{v}}$ in f .

We have the following reductions between these problems [4], [3]:

$$\text{EqSLP} \leq_m \text{DegSLP} \leq_r \text{CoeffSLP} \leq \mathbf{PP}^{\mathbf{NP}} \approx \mathbf{P}^{\mathbf{PP}} \in C_2$$

where \leq_m denotes a polynomial-time many-one reduction and \leq_r denotes a randomized polynomial-time reduction.

It is known that CoeffSLP is **#P**-hard (see for instance [3]) and from the proof of [4, Theorem 4.1] it can be shown to be in **FP^{#P}**. Meanwhile **DegSLP** is in **coRP^{PP}** [4, Theorem 1.5], but is not known to be **NP**-hard.

C. Formal Power Series and Algebraic Equation Systems

The ring of multivariate *formal power series* with integer coefficients over indeterminates $X = (x_1, \dots, x_k)$ is defined by

$$\mathbb{Z}[[X]] := \left\{ \sum_{\mathbf{v} \in \mathbb{N}^k} a_{\mathbf{v}} X^{\mathbf{v}} \mid a_{\mathbf{v}} \in \mathbb{Z} \right\}.$$

We denote by \mathfrak{m} the ideal of the ring $\mathbb{Z}[[X]]$ generated by x_1, \dots, x_k , i.e., the ideal of all power series with no constant term. For instance, $x_2 + x_3^2 \in \mathfrak{m}$, but $1 + x_1 \notin \mathfrak{m}$. In language theory such series are traditionally called *quasiregular* [1, Chapter IV]. We note that the units of $\mathbb{Z}[[X]]$, that is, the elements with multiplicative inverses, are those of the form $\pm 1 + f$ where $f \in \mathfrak{m}$.

A series $A \in \mathbb{Z}[[X]]$ is said to be *algebraic* if there exist polynomials $p_0, \dots, p_d \in \mathbb{Z}[X]$, not all zero, such that

$$p_0 + p_1 A + p_2 A^2 + \dots + p_d A^d = 0.$$

²Our usage here is non-standard in that EqSLP typically refers to the problem of determining zeroness of an arithmetic circuit that represents an integer. However the different versions of the problem are irreducible, so the distinction is not significant.

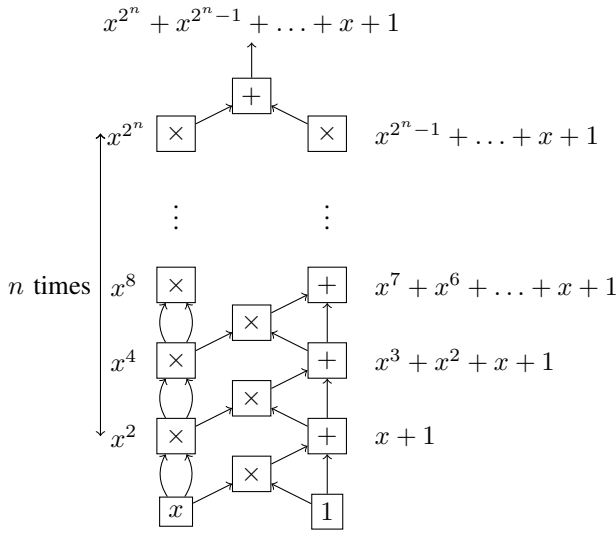


Fig. 1. An arithmetic circuit representing the polynomial $\sum_{i=0}^{2^n-1} x^i$ of size $O(n)$. This is a special case of Proposition 6 applied to $m = 2^n$.

We call the polynomial $\sum_{n=0}^d p_n y^n \in \mathbb{Z}[X][y]$ an *annihilating polynomial* of A .

Let $Y = (y_1, \dots, y_\ell)$ be a tuple of variables. A *polynomial equation system* S over the indeterminates X and variables Y consists of a collection of ℓ equations

$$y_i = P_i \quad (i \in \{1, \dots, \ell\})$$

where $P_i \in \mathbb{Z}[X][Y]$. In computational problems we assume that the polynomials P_i are presented as lists of monomials and integers are written in binary. The *size* of S is the sum of sizes of its polynomials P_1, \dots, P_ℓ ; in particular, it is at least the number k of indeterminates and ℓ of variables. In case the coefficients of the P_i are all polynomials with nonnegative integer coefficients from \mathbb{N} , we say that S is *defined over* \mathbb{N} .

A vector $\mathbf{A} \in \mathbb{Z}[[X]]^\ell$ of formal power series evaluating each polynomial P_i at \mathbf{A} yields an identity $A_i = P_i(\mathbf{A})$ of formal power series.

We introduce a condition guaranteeing unique solutions of a certain kind. We say that the system S is *proper* if for each polynomial $P_i = \sum a_v Y^v$, we have $a_v \in \mathfrak{m} \cap \mathbb{Z}[X]$ whenever $|v| \leq 1$. In other words, the coefficients of monomials of total degree at most one are in the ideal \mathfrak{m} .

Inductively define the *approximating sequence* $\mathbf{A}^{(0)}, \mathbf{A}^{(1)}, \dots$ in $\mathbb{Z}[[X]]^\ell$ by $\mathbf{A}^{(0)} := \mathbf{0}$ and

$$\mathbf{A}^{(n+1)} := (P_1(\mathbf{A}^{(n)}), \dots, P_\ell(\mathbf{A}^{(n)})) \quad (1)$$

for all $n \in \mathbb{N}$. When applied to a proper system, every component in this sequence converges pointwise to a vector of power series $\mathbf{A} \in \mathbb{Z}[[X]]^\ell$ with respect to the product topology of $\mathbb{Z}[[X]]$. This vector \mathbf{A} is a solution of the system, called the *strong solution* [1, Section IV.1, Theorem 1.1]. Moreover, the strong solution is the unique quasiregular solution, and each component thereof is algebraic [7]. We say that the first component of this unique solution is the *formal power series*

computed by S . More generally, if for an arbitrary system S the approximating sequence (1) converges to \mathbf{A} , then the quasiregular part of \mathbf{A} (which is obtained by omitting the term of degree zero) is the solution of a proper polynomial system [2].

Example 1. The polynomial equation $y = x + x^2 - 2xy + y^2$ is proper with two solutions x and $1 + x$, of which only x is quasiregular. The approximating sequence in this case is $A_0 = 0$ and $A_n = x + x^{2^n}$, which can be proved by induction. In particular, A_n converges to x as required.

The generating function $C(x)$ of the Catalan numbers is a solution of the polynomial equation $y = 1 + xy^2$, which is not proper. However, its quasiregular part $C(x) - 1$ is the solution of the proper system $y = x + 2xy + xy^2$.

In analogy with the problems EqSLP, DegSLP, and CoeffSLP for algebraic circuits, we consider in this paper the following problems on algebraic power series.

- **EqAlg:** Given a proper system of polynomial equations with unique quasiregular solution \mathbf{A} , decide whether the first component A_1 of \mathbf{A} is the zero power series.
- **FinAlg:** Given a positive integer d and a proper system of polynomial equations with unique quasiregular solution \mathbf{A} , decide whether the total degree of the first component A_1 of \mathbf{A} is finite.
- **CoeffAlg:** Given a multi-index \mathbf{v} , a prime p (both encoded in binary), and a proper system of polynomial equations with unique quasiregular solution \mathbf{A} , compute the residue module p of the coefficient of the monomial $X^{\mathbf{v}}$ in the first component A_1 of \mathbf{A} .

D. Hensel's Lemma

Let R be a commutative ring with unity. A *valuation* on R is a map $v : R \rightarrow \mathbb{N} \cup \{\infty\}$ such that for all $x, y \in R$ we have:

- 1) $v(x) = \infty$ iff $x = 0$,
- 2) $v(x + y) \geq \min\{v(x), v(y)\}$,
- 3) $v(xy) = v(x) + v(y)$.

It is easy to check that, if v is a valuation on R , then the function $d(x, y) := 2^{-v(x-y)}$ (with the convention that $2^{-\infty} = 0$) defines an ultrametric on R . We say that R is *complete* with respect to v if it is a complete metric space with respect to d (in the standard sense).

We now state a version of Hensel's Lemma that is convenient for our purposes. This combines the multivariate Hensel Lemma found in [19, Section 4.6, Theorem 2] and [20, Exercise 7.26] with an assertion of quadratic convergence. We will use Hensel's Lemma in Sec. III-B to define a sequence of rational approximations converging to the quasiregular solution of a proper system of polynomial equations. The proof is a straightforward adaptation of classical arguments from the literature, however we include a proof for the convenience of the reader.

Assume that the ring R is complete with respect to a valuation v . Let f_1, \dots, f_ℓ lie in the polynomial ring $R[Y]$,

Could have assumed right at the beginning that the Jacobian matrix is nilpotent.

where $Y = (y_1, \dots, y_\ell)$ is a tuple of distinct indeterminates. Recall that the *derivative matrix* of $\mathbf{f} = (f_1, \dots, f_\ell)$ is

$$D\mathbf{f} := \left(\frac{\partial f_i}{\partial y_j} \right)_{1 \leq i, j \leq \ell} \in R[Y]^{\ell \times \ell} \quad (2)$$

and its *Jacobian* is $J_{\mathbf{f}} := \det(D\mathbf{f}) \in R[Y]$.

For a positive integer ℓ , we extend v to a map $v : R^\ell \rightarrow \mathbb{N}$ by writing $v(x_1, \dots, x_\ell) := \min\{v(x_i) : 1 \leq i \leq \ell\}$. Note that the extension of v to R^ℓ is not in general a valuation.

Theorem 8 (Hensel's Lemma). *Let $\mathbf{a} \in R^\ell$ be such that $v(\mathbf{f}(\mathbf{a})) > 0$ and $J_{\mathbf{f}}(\mathbf{a})$ is a unit in R . Consider the sequence $(\mathbf{a}_n)_{n=0}^\infty$, defined inductively by $\mathbf{a}_0 = \mathbf{a}$ and*

$$\mathbf{a}_{n+1} = \mathbf{a}_n - (D\mathbf{f}(\mathbf{a}_n))^{-1} \mathbf{f}(\mathbf{a}_n) \quad (n \in \mathbb{N}). \quad (3)$$

Then $(\mathbf{a}_n)_{n=0}^\infty$ is a well-defined sequence in R^ℓ and there exists $\alpha \in R^\ell$ such that $\mathbf{f}(\alpha) = 0$ and $v(\alpha - \mathbf{a}_n) \geq 2^n$ for all $n \in \mathbb{N}$. The statement of well-definedness refers to the claim that the matrix $D\mathbf{f}(\mathbf{a}_n)$ is invertible for all n , which amounts to the fact that $J_{\mathbf{f}}(\mathbf{a}_n)$ is a unit of R for all n .

Proof. We will need the following elementary fact about complete rings (see, for example, [21, Lemma 2.6]): If $x \in R$ is a unit and $v(x - y) > 0$ then y is also a unit.

We will show by induction on $n \in \mathbb{N}$ that the sequence $(\mathbf{a}_n)_{n=0}^\infty$ is well-defined and satisfies the following for all n :

- 1) $v(\mathbf{f}(\mathbf{a}_n)) \geq 2^n$,
- 2) $v(\mathbf{a}_n - \mathbf{a}_{n-1}) \geq 2^{n-1}$ if $n > 0$,
- 3) $J_{\mathbf{f}}(\mathbf{a}_n)$ is a unit.

Suppose first that $n = 0$. Then, Item 2 holds vacuously, while, by the choice $\mathbf{a}_0 = \mathbf{a}$, Items 1 and 3 are hypotheses of the theorem.

For the induction step, by the multivariate Taylor's Theorem, for $\varepsilon = (\varepsilon_1, \dots, \varepsilon_\ell) \in R^\ell$ it holds that

$$\mathbf{f}(\mathbf{a}_n + \varepsilon) = \mathbf{f}(\mathbf{a}_n) + ((D\mathbf{f})(\mathbf{a}_n))\varepsilon + \sum_{1 \leq i, j \leq \ell} \varepsilon_i \varepsilon_j \mathbf{c}_{ij}, \quad (4)$$

where $\mathbf{c}_{ij} \in R^\ell$ for all $1 \leq i, j \leq \ell$. By Item 3 the matrix $(D\mathbf{f})(\mathbf{a}_n)$ has an inverse that lies in $R^{\ell \times \ell}$. Thus we may put $\varepsilon := -((D\mathbf{f})(\mathbf{a}_n))^{-1} \mathbf{f}(\mathbf{a}_n) \in R^\ell$, in which case $v(\varepsilon) \geq v(\mathbf{f}(\mathbf{a}_n)) \geq 2^n$. The first inequality follows from the general fact that $v(A \cdot \mathbf{b}) \geq v(\mathbf{b})$ for every matrix $A \in R^\ell$ and vector $\mathbf{b} \in R^\ell$. By Equations (3) and (4),

$$\mathbf{f}(\mathbf{a}_{n+1}) = \mathbf{f}(\mathbf{a}_n + \varepsilon) = \sum_{1 \leq i, j \leq \ell} \varepsilon_i \varepsilon_j \mathbf{c}_{ij}.$$

We conclude that $v(\mathbf{f}(\mathbf{a}_{n+1})) \geq 2^{n+1}$, as required in Item 1. We also have $v(\mathbf{a}_{n+1} - \mathbf{a}_n) = v(\varepsilon) \geq 2^n$, as required in Item 2.

Regarding Item 3, we will use the following claim (proved in Appendix A).

Claim 9. *For a univariate polynomial $p(x) \in R[x]$ and two elements $a, b \in R$, $v(p(a) - p(b)) \geq v(a - b)$.*

Since $J_{\mathbf{f}}$ is a polynomial in $R[Y]$, by Claim 9 componentwise, it follows that

$$v(J_{\mathbf{f}}(\mathbf{a}_{n+1}) - J_{\mathbf{f}}(\mathbf{a}_n)) \geq v(\mathbf{a}_{n+1} - \mathbf{a}_n) > 0$$

and hence, since $J_{\mathbf{f}}(\mathbf{a}_n)$ is a unit, by the observation right at the start of the proof, $J_{\mathbf{f}}(\mathbf{a}_{n+1})$ is a unit as well. This establishes Item 3 and the induction is complete. Note also that this implies that the sequence $(\mathbf{a}_n)_{n=0}^\infty$ is well-defined, and by construction, for all n , $\mathbf{a}_n \in R^\ell$.

By Item 2 we have $v(\mathbf{a}_m - \mathbf{a}_n) \geq 2^n$ for all $m > n$. Hence the sequence $(\mathbf{a}_n)_{n=0}^\infty$ is Cauchy and so, by completeness of R , it converges to a limit $\alpha \in R^\ell$. But by Item 1 and continuity of \mathbf{f} we have $\mathbf{f}(\alpha) = 0$. Furthermore, by continuity of v we have $v(\alpha - \mathbf{a}_n) \geq 2^n$. \square

Remark 10. *One often finds formulations of Hensel's Lemma that require the ring R to be a valuation ring, i.e., the subring of a valued field comprising all elements having non-negative valuation. Examples of valuation rings are the p -adic integers \mathbb{Z}_p (which is the valuation ring of the field \mathbb{Q}_p) and the ring $F[[x]]$ of univariate power series over a field F (which is the valuation ring of the field $F((x))$ of Laurent series). However, as we note in Sec. III, Theorem 8 applies to the ring of multivariate power series $\mathbb{Z}[[X]]$, which is not a valuation ring.*

III. COMPUTATIONAL COMPLEXITY OF CoeffAlg

Recall that an instance of the CoeffAlg problem comprises a proper polynomial equation system computing a formal power series A , a monomial X^v , and a prime p . The problem asks to compute the residue modulo p of the coefficient of X^v in A . In analysing the complexity of the problem we assume that all integers are represented in binary. In this section we prove the following theorem:

Theorem 1. *CoeffAlg is equivalent under polynomial time reductions to CoeffSLP and hence is $\#P$ -hard and in $\mathbf{FP}^{\#P}$.*

The proof is spread across several subsections. Throughout we work with a tuple $X = (x_1, \dots, x_k)$ of commuting indeterminates and a proper polynomial equation system \mathcal{S} over the indeterminates X and variables $Y = (y_1, \dots, y_\ell)$ —forming part of the input to the CoeffAlg problem. We let s denote the length of the description of \mathcal{S} .

Let $\mathbf{A} = (A_1, \dots, A_\ell) \in \mathbb{Z}[[X]]^\ell$ be the unique quasiregular solution of \mathcal{S} in formal power series. Roughly speaking our reduction of CoeffAlg to CoeffSLP involves constructing a sufficiently close polynomial approximation of A_1 that admits an efficiently computable representation as a circuit. To do this we will apply Hensel's Lemma to the power series ring $\mathbb{Z}[[X]]$.

A. A Valuation on the Ring of Power Series

In the following we denote by R the ring of formal power series $\mathbb{Z}[[X]]$ and by R_0 the subring $\mathbb{Z}[X]$ of polynomials. Recall that \mathfrak{m} is the ideal in R generated by x_1, \dots, x_k . Given $g \in \mathfrak{m}$, the element $1 - g$ is a unit in R , having inverse $\sum_{n=0}^\infty g^n$. (The latter sum converges as a power series by

the prefix is correct but the trailing terms need not be 0

virtue of the fact that g is in \mathfrak{m} .) Indeed, the units in R are precisely those elements f such that $\pm f$ has the above form. An element of R is said to be *rational* if it has the form fg^{-1} , where $f, g \in R_0$ and g is a unit.³

We define a map $\text{ord} : R \rightarrow \mathbb{N} \cup \{\infty\}$ by $\text{ord}(0) := \infty$ and otherwise

$$\text{ord}\left(\sum_{\mathbf{v} \in \mathbb{N}^k} a_{\mathbf{v}} X^{\mathbf{v}}\right) := \min\{|\mathbf{v}| : a_{\mathbf{v}} \neq 0\}.$$

It is easily seen that the map ord is a valuation: for all $f, g \in R$

$$\text{ord}(f + g) \geq \min\{\text{ord}(f), \text{ord}(g)\},$$

and

$$\text{ord}(fg) = \text{ord}(f) + \text{ord}(g).$$

Since elements with a strictly positive valuation are precisely the elements without a constant term, we recover the ideal \mathfrak{m} as the set $\{f \in R : \text{ord}(f) > 0\}$. *including $\text{ord}(f) = \infty$*

It is standard that R is a complete ring with respect to this valuation; see [20, Section 7.1]. For $\ell \in \mathbb{N}$, we extend the map ord to vectors R^ℓ of power series by, for all $\mathbf{B} \in R^\ell$,

$$\text{ord}(\mathbf{B}) := \min\{\text{ord}(B_i) : 1 \leq i \leq \ell\} \in \mathbb{N} \cup \{\infty\}$$

B. Approximation by Rational Series

For our fixed polynomial system \mathcal{S} , given by the equations

$$y_i = P_i \quad (i \in \{1, \dots, \ell\})$$

we write $f_i := y_i - P_i \in R_0[Y]$. We denote by \mathbf{f} the vector of polynomials

$$\mathbf{f} = (f_1, \dots, f_\ell).$$

Clearly $\mathbf{B} \in R^\ell$ is a solution of \mathcal{S} if and only if $\mathbf{f}(\mathbf{B}) = 0$. Since \mathcal{S} is proper it follows that \mathbf{f} has a unique quasiregular zero, namely the unique quasiregular solution \mathbf{A} of \mathcal{S} .

Our goal is to apply Theorem 8—the multivariate Hensel's Lemma—to define a sequence of approximations of the unique quasiregular zero \mathbf{A} of \mathbf{f} . We have already established that R is complete with respect to the valuation ord . To apply Hensel's Lemma it remains to find a suitable initial value to start the iteration, namely, $\mathbf{a} \in R^\ell$ such that $J_{\mathbf{f}}(\mathbf{a})$ is a unit in R and $\text{ord}(\mathbf{f}(\mathbf{a})) > 0$. The following two claims show that any $\mathbf{a} \in \mathfrak{m}^\ell$ will do. Recall that $1 + \mathfrak{m}$ denotes the set of elements of the form $1 + f$ with $f \in \mathfrak{m}$.

Claim 11. $J_{\mathbf{f}}(\mathbf{a}) \in 1 + \mathfrak{m}$ for all $\mathbf{a} \in \mathfrak{m}^\ell$.

We give a sketch proof of the claim; see Appendix B for the full proof. Assume that $\mathbf{a} \in \mathfrak{m}^\ell$. Using the fact that \mathcal{S} is proper, one can show that the diagonal entries of the derivative matrix $D\mathbf{f}(\mathbf{a})$ all lie in $1 + \mathfrak{m}$ whereas the off-diagonal entries all lie in \mathfrak{m} . In other words, the matrix $D\mathbf{f}(\mathbf{a})$ is elementwise congruent to the identity matrix modulo \mathfrak{m} .

³Since \mathbb{Z} is a so-called Fatou ring [22, Chapter 7], the rational elements of R according to the above definition are precisely those lying in $\mathbb{Q}(X) \cap R$, where $\mathbb{Q}(X)$ is the field of rational functions over indeterminates X .

Since the determinant of a matrix is a polynomial function of its entries, $J_{\mathbf{f}}(\mathbf{a}) \in 1 + \mathfrak{m}$.

The fact that $\text{ord}(\mathbf{f}(\mathbf{a})) > 0$ for all $\mathbf{a} \in \mathfrak{m}^\ell$ follows immediately from

Claim 12. For all $\mathbf{a} \in \mathfrak{m}^\ell$ we have $\mathbf{f}(\mathbf{a}) \in \mathfrak{m}^\ell$.

The claim holds by observing that, since \mathcal{S} is proper, the coefficient of Y^0 in $f_i \in R_0[Y]$ lies in \mathfrak{m} . Again, see Appendix B for details.

Claims 11 and 12 imply that choosing \mathbf{a} to be $\mathbf{0} \in R^\ell$ satisfies the assumptions of Theorem 8. Now define a sequence $(\mathbf{a}_n)_{n=0}^\infty$ by $\mathbf{a}_0 = \mathbf{0}$ and

$$\begin{aligned} \mathbf{a}_{n+1} &= \mathbf{a}_n - (D\mathbf{f})(\mathbf{a}_n)^{-1} \mathbf{f}(\mathbf{a}_n) \\ &= \mathbf{a}_n - \frac{1}{J_{\mathbf{f}}(\mathbf{a}_n)} \text{Adj}(D\mathbf{f}(\mathbf{a}_n)) \mathbf{f}(\mathbf{a}_n), \end{aligned} \quad (6)$$

where the entries of $D\mathbf{f}$, the derivative matrix of \mathbf{f} , are in $R_0[Y]$, $J_{\mathbf{f}}$ is the determinant of $D\mathbf{f}$, as defined in Equation (2). Here, $\text{Adj}(D\mathbf{f}(\mathbf{a}_n))$ is the adjugate matrix whose entries are just the cofactors of $D\mathbf{f}(\mathbf{a}_n)$. Applying Theorem 8 to \mathbf{f} , for all $n \in \mathbb{N}$ we have

$$\text{ord}(\mathbf{A} - \mathbf{a}_n) \geq 2^n. \quad (7)$$

The sequence $(\mathbf{a}_n)_{n=0}^\infty$ is the desired approximating sequence of \mathbf{A} .

Note that by Theorem 8(3), the Jacobian $J_{\mathbf{f}}(\mathbf{a}_n)$ is a unit in R for all $n \in \mathbb{N}$. Since the rational elements of R (i.e., the rational power series) form a ring, it follows by a straightforward induction on n , using the recurrence (6), that each \mathbf{a}_n is a vector of rational elements in R . Consequently, for all $i \in \{1, \dots, \ell\}$ and $n \in \mathbb{N}$, the component $\mathbf{a}_{n,i}$ can be written for some polynomials $g_{n,i}, h_{n,i} \in R_0$ as

$$\mathbf{a}_{n,i} = \frac{g_{n,i}}{1 - h_{n,i}}, \quad \text{with } \text{ord}(h_{n,i}) \geq 1. \quad (8)$$

Example 2. Consider again the polynomial system $y = x + x^2 - 2xy + y^2$, introduced in Example 1, with the unique quasiregular solution x . Write $f(y) = y - x - x^2 + 2xy - y^2$ for the polynomial defining the system and $D_f(y) = 1 + 2x - 2y$ for its derivative with respect to y . Then following the iterative procedure given by Hensel's Lemma, for all $n \in \mathbb{N}$, we have

$$a_{n+1} = a_n - \frac{f(a_n)}{D_f(a_n)} = \frac{-a_n^2 + x^2 + x}{1 + 2x - 2a_n}. \quad (9)$$

Starting the iteration with $a_0 = 0$, we prove by induction that the term a_n , for $n \geq 1$, has the following closed-form expression:

$$a_n = x - \frac{x^{2^n}}{(x+1)^{2^n} - x^{2^n}}. \quad (10)$$

The inductive proof is detailed in Appendix C. We note that $\text{ord}(a_n - x) = 2^n$, matching the bound in Theorem 8.

C. Approximation by Polynomials

In the previous section, we have shown how to approximate the quasiregular solution A by rational power series with good convergence speed, see (7). In this section we achieve the same convergence by approximating *even by polynomials*. This will allow us to implement the polynomial approximants with algebraic circuits in Sec. III-D. To this end, we define a sequence $\tilde{\mathbf{a}}_n \in R_0^\ell$ of vectors of polynomials that satisfies the following convergence bound for all n (see Equation (7)):

$$\text{ord}(A_i - \tilde{\mathbf{a}}_{n,i}) \geq 2^n \quad (11)$$

where $\tilde{\mathbf{a}}_{n,i}$ is the i -th component of $\tilde{\mathbf{a}}$. The definition of $\tilde{\mathbf{a}}_{n,i}$ uses a classical technique of division elimination by Strassen [23]. While Strassen obtains a polynomial written as a rational function, we are computing a polynomial approximation of the power series defined by a rational function.

Given the representation of $\mathbf{a}_{n,i}$ in Equation (8), we have

$$\mathbf{a}_{n,i} = g_{n,i} \sum_{j=0}^{\infty} h_{n,i}^j.$$

A polynomial approximant $\tilde{\mathbf{a}}_{n,i}$ can then be obtained by truncating the above infinite sum to the first 2^n terms. We thus write

$$\tilde{\mathbf{a}}_{n,i} := g_{n,i} \sum_{j=0}^{2^n-1} h_{n,i}^j. \quad (12)$$

Then we have

$$\begin{aligned} \text{ord}(\mathbf{a}_{n,i} - \tilde{\mathbf{a}}_{n,i}) &= \text{ord}(g_{n,i} \sum_{j=2^n}^{\infty} h_{n,i}^j) \\ &= \text{ord}(g_{n,i}) + \text{ord}\left(\sum_{j=2^n}^{\infty} h_{n,i}^j\right) \\ &\geq 2^n \quad (\text{since } \text{ord}(h_{n,i}) \geq 1) \end{aligned}$$

The desired bound in Equation (11) now follows from (7) and the strong triangle inequality (Property 2 of valuations in Sec. II-D):

$$\begin{aligned} \text{ord}(A_i - \tilde{\mathbf{a}}_{n,i}) &= \text{ord}(A_i - \mathbf{a}_{n,i} + \mathbf{a}_{n,i} - \tilde{\mathbf{a}}_{n,i}) \\ &\geq \min\{\text{ord}(A_i - \mathbf{a}_{n,i}), \text{ord}(\mathbf{a}_{n,i} - \tilde{\mathbf{a}}_{n,i})\} \\ &\geq 2^n. \end{aligned}$$

D. Approximation by Circuits

In this subsection we show that for all $n \in \mathbb{N}$ the polynomial approximant $\tilde{\mathbf{a}}_n$ of \mathbf{A} can be represented by a circuit that can moreover be computed in time polynomial in n and the size s of the equation system \mathcal{S} . We construct the circuit in two steps. First we show how to construct circuits for the two polynomials $g_{n,i}, h_{n,i} \in R_0$ in Equation (8) representing $\mathbf{a}_{n,i}$.

Claim 13. *There is an algorithm that, given an equation system \mathcal{S} , $i \in \{1, \dots, \ell\}$, and $n \in \mathbb{N}$, produces circuits $C_{n,i}, D_{n,i}$ representing polynomials $g_{n,i}, h_{n,i} \in R_0$ respectively, as in Equation (8). The algorithm runs in $\text{poly}(s, n)$ time and hence $C_{n,i}, D_{n,i}$ have size $\text{poly}(s, n)$.*

Next, we follow the procedure described in Section III-C to construct a circuit for the approximants $\tilde{\mathbf{a}}_{n,i}$ obtained from $g_{n,i}$ and $h_{n,i}$. Since we will need these approximants just for the first component $i = 1$, we introduce this specialisation already in the next claim.

Claim 14. *There is an algorithm that, given an equation system \mathcal{S} and $n \in \mathbb{N}$, produces circuit E_n representing the polynomial $\tilde{\mathbf{a}}_{n,1}$, defined in Equation (12) for $i = 1$. The algorithm runs in $\text{poly}(s, n)$ time, and E_n has size $\text{poly}(s, n)$.*

E. The upper and lower complexity bounds

We are now ready to conclude the proof of Theorem 1. By Equation (11) and Claim 14, for our fixed proper polynomial system \mathcal{S} , for all $n \in \mathbb{N}$, we compute a circuit E_n representing a polynomial approximant of the formal power series solution A computed by \mathcal{S} . Indeed, the polynomial represented by E_n agrees with A in all monomials with total degree at most 2^n . Moreover, there is an algorithm computing E_n in $\text{poly}(s, n)$ time, and E_n has size $\text{poly}(s, n)$. It remains to observe that for the input monomial $X^{\mathbf{v}}$ and prime p , the residue modulo p of the coefficient of $X^{\mathbf{v}}$ in A and $E_{\log(|\mathbf{v}|)+1}$ agree. This yields a polynomial-time reduction from CoeffAlg to CoeffSLP.

For the converse direction, note that CoeffSLP is not trivially subsumed by CoeffAlg as the latter requires the input system of equations to be proper. The following claim, proven in Appendix B, outlines a straightforward reduction from CoeffSLP to CoeffAlg.

Claim 15. *There is a polynomial-time reduction from CoeffSLP to CoeffAlg.*

IV. COMPLEXITY OF EqAlg AND FinAlg

In this section we establish complexity upper bounds for EqAlg and FinAlg. A key ingredient behind these results is a new singly exponential upper bound on the degree of an annihilating polynomial of the strong solution of a proper polynomial system.

There are two different algorithms that given a proper polynomial system \mathcal{S} compute an annihilating polynomial of its strong solution. The algorithm of Kuich and Salomaa [2, Section 16] is based on multiresultants and polynomial factorisation, whereas the algorithm of Panholzer [7] uses Gröbner bases. It is noted in [7, Example 9] that one cannot always obtain an annihilating polynomial of the strong solution merely by performing quantifier elimination on the system \mathcal{S} over the first-order theory of algebraically closed fields. This is because the elimination ideal may be trivial and does not provide any information on the strong solution. In such a case further work is needed to isolate the strong solution, such as decomposing the variety of all solutions of \mathcal{S} into its irreducible components as in [7].

In this section we take an alternative approach. We observe that the strong solution is first-order definable in the ordered field of Puiseux series and we use standard quantifier elimination results for real closed fields to compute an annihilating

polynomial of the strong solution. In particular, we obtain a singly exponential (in the size of S) upper bound on the degree of an annihilating polynomial. Paper [6] states a doubly exponential upper bound on the degree of an annihilating polynomial, based on an analysis of the Kuich-Salomaa algorithm.

A. Real Closed Fields and Puiseux Series

We work with the first-order theory of real closed fields over the language of ordered rings with constant symbols for 0 and 1 [24, Chapter 2]. Recall that a model of this theory is an ordered field in which the intermediate value theorem holds for all polynomials (such as the field of real numbers or the field of Puiseux series with real coefficients, defined below). Atomic formulas have the form $P(x_1, \dots, x_n) \sim 0$, where $P \in \mathbb{Z}[x_1, \dots, x_n]$ and $\sim \in \{<, =\}$. We say that a formula Φ is *built over a set of polynomials* \mathcal{P} if every polynomial mentioned in Φ lies in \mathcal{P} . It is well-known that the theory of real closed fields admits quantifier elimination. Here we will use the following quantitative formulation of quantifier elimination, which is a specialisation of [24, Theorem 14.16].

Theorem 16. *Let \mathcal{P} be a set of s polynomials, each of degree at most d and having coefficients of bit-size at most τ . Given tuples $X = (x_1, \dots, x_{k_1})$, $Y = (y_1, \dots, y_{k_2})$ and $Z = (z_1, \dots, z_{k_3})$ of first-order variables, consider the formula*

$$\Phi(X) := \exists Y \forall Z \Psi(X, Y, Z),$$

where $\Psi(X, Y, Z)$ is a quantifier-free formula built over \mathcal{P} . Then there exists an equivalent⁴ quantifier-free formula $\Phi'(X)$ that is built over a set of polynomials having degree bounded by $d^{ck_2k_3}$ and coefficients of bit size bounded by $\tau d^{ck_1k_2k_3}$ for some absolute constant c .

Let F be a field and $X = (x_1, \dots, x_k)$ a tuple of commuting indeterminates. A *Puiseux series* with coefficients in F and variables X is a formal series

$$f := \sum_{\alpha \in \mathbb{Q}^k} c_\alpha X^\alpha \quad (13)$$

whose *support* $S := \{\alpha \in \mathbb{Q}^k : c_\alpha \neq 0\}$ is well-ordered with respect to the lexicographic order on \mathbb{Q}^k and also satisfies $S \subseteq \frac{1}{q}\mathbb{Z}$ for some positive integer q . The collection of Puiseux series over F forms a field $F\{\{X\}\}$ with the obvious definitions of sum and product. (Note that the product is well-defined thanks to the well-foundedness of the support.) In case F is an ordered field we can lift the order on F to $F\{\{X\}\}$ by declaring that a non-zero series f as in (13) is positive just in case $c_{\alpha_0} > 0$, where α_0 is the least element of the support of f (w.r.t. the lexicographic order on \mathbb{Q}^k). We then declare $f < g$ just in case $g - f$ is positive. If F is a real closed field then $F\{\{X\}\}$ is a real closed field under the above order [24, Theorem 2.91].⁵

⁴Equivalent over every real closed field.

⁵This result is usually stated in the case of univariate Puiseux series, but the multivariate version follows by induction, since $F\{\{X\}\} := F\{\{x_1\}\}\{\{x_2\}\} \cdots \{\{x_k\}\}$.

B. The Strong Solution is Algebraic

For the rest of this section let $X = (x_1, \dots, x_k)$ be tuple of commuting indeterminates. Consider a proper polynomial system S over a set of variables $Y = (y_1, \dots, y_\ell)$, given by equations

$$y_i = P_i \quad (i = 1, \dots, \ell) \quad (14)$$

where $P_i \in \mathbb{Z}[X][Y]$ for all $i \in \{1, \dots, \ell\}$. A solution of S in Puiseux series is a tuple $(A_1, \dots, A_\ell) \in \mathbb{R}\{\{X\}\}^\ell$ such that evaluating each polynomial P_i at (A_1, \dots, A_ℓ) yields an identity $A_i = P_i(A_1, \dots, A_\ell)$ of Puiseux series. We say that such a solution is *non-negative* if $A_i \geq 0$ for all $i = 1, \dots, \ell$. In other words, the first coefficient of each of the A_i 's is strictly positive, if it exists.

Proposition 17. *The strong solution of a proper polynomial system that is defined over \mathbb{N} is the least non-negative solution among Puiseux series.*

Proof. Consider a proper polynomial system S , as shown in (14). Assume that S is defined over \mathbb{N} . Let $B \in \mathbb{R}\{\{X\}\}^\ell$ be a non-negative solution of S . Recall the approximating sequence $A^{(0)}, A^{(1)}, \dots$ of the strong solution A of S , which is defined inductively by $A^{(0)} := 0$ and, for all $n \in \mathbb{N}$,

$$A^{(n+1)} := (P_1(A^{(n)}), \dots, P_\ell(A^{(n)})).$$

Since the integer coefficients in each polynomial P_i in S are non-negative, P_i defines a monotone function from $\mathbb{R}\{\{X\}\}^\ell$ to $\mathbb{R}\{\{X\}\}$. Now we have $0 \leq B$ by assumption. Moreover if we inductively assume that $A^{(j)} \leq B$ then, since each polynomial P_i has coefficients in $\mathbb{N}[X] \subseteq \mathbb{Z}[X]$, we have

$$\begin{aligned} A^{(j+1)} &= (P_1(A^{(j)}), \dots, P_\ell(A^{(j)})) \\ &\leq (P_1(B), \dots, P_\ell(B)) \\ &= B. \end{aligned}$$

We conclude that $A^{(j)} \leq B$ for all j , and hence $A \leq B$. \square

Theorem 18. *There is an absolute constant c with the following property. Given a strong solution $A \in \mathbb{Z}[[X]]$ of a proper polynomial system on ℓ variables and involving polynomials of total degree at most d , there is a polynomial $P \in \mathbb{Z}[X][y]$ of total degree at most $d^{c\ell^2}$ such that $P(A) = 0$.*

Proof. Let us prove the theorem first in the special case that is defined over \mathbb{N} . Assume that S is as displayed in Equation (14). Consider the following first-order formula $\Phi(U, v)$ in free variables $U = (u_1, \dots, u_k)$ and v , involving also bound variables $Y = (y_1, \dots, y_\ell)$ and $Z = (z_1, \dots, z_\ell)$:

$$\begin{aligned} \Phi(U, v) := & \exists Y \forall Z \\ & \left(Y \geq 0 \wedge Y = (P_1(U, Y), \dots, P_\ell(U, Y)) \right. \\ & \wedge (Z \geq 0 \wedge Z = (P_1(U, Z), \dots, P_\ell(U, Z))) \\ & \left. \rightarrow Y \leq Z \right) \wedge v = y_1 \end{aligned}$$

Intuitively this formula expresses that Y is the least non-negative solution of S .

Consider a variable assignment $\alpha : U \cup \{v\} \rightarrow \mathbb{R}\{\{X\}\}$ satisfying $\alpha(u_i) = x_i$ for $i = 1, \dots, k$. By Proposition 17 the unique value of $\alpha(v)$ such that $\Phi(U, v)$ is satisfied by α is the strong solution of S .

By Theorem 16 there is a quantifier-free formula $\Phi'(U, v)$ that has the same set of satisfying assignments as Φ over any real closed field, and in particular over $\mathbb{R}\{\{X\}\}$, and that is moreover built over a family $\mathcal{P} \subseteq \mathbb{Z}[U][v]$ of polynomials of total degree at most $d^{c\ell^2}$. Consider again the assignment α introduced above. By the uniqueness of the strong solution there must be an inequality $P \geq 0$ in Φ' such that the equality $P = 0$ holds under assignment α . (Indeed if none of the inequalities in Φ' were tight under the assignment α then a suitably small perturbation of the value $\alpha(v)$ would yield second satisfying assignment of Ψ' .) In other words, there is a polynomial $P \in \mathcal{P}$ such that the equation $P(X, A) = 0$ holds in $\mathbb{R}\{\{X\}\}$.

It remains to handle the case that A is a solution of a general polynomial system S with ℓ variables and involving polynomials of total degree at most d . Here we can write A as the difference $A = A^{(1)} - A^{(2)}$ of two series that are components of the solution of a proper system that is defined over \mathbb{N} and involves 2ℓ variables and polynomials of degree at most d [1, Section IV.2, Theorem 2.4]. The reasoning above shows that the two series $A^{(1)}$ and $A^{(2)}$ are definable over $\mathbb{R}\{\{X\}\}$ by $\exists^*\forall^*$ formulas and hence the series A is also definable by such a formula. The rest of the proof now follows as in the case of a single system that is defined over \mathbb{N} . \square

C. Bounds on the Order and Degree

The following proposition relates the order and degree (when finite) of an algebraic power series in $\mathbb{Z}[[X]]$ to the degree of its annihilating polynomial. Given $f = \sum_{v \in \mathbb{N}^k} \alpha_v X^v \in \mathbb{Z}[[X]]$ and $D \in \mathbb{N}$, define

$$\text{tail}_D(f) := \sum_{v \in \mathbb{N}^k, |v| > D} \alpha_v X^v$$

to be the series obtained by deleting all monomials of total degree at most D .

Proposition 19. *Let $f = \sum_{v \in \mathbb{N}^k} \alpha_v X^v \in \mathbb{Z}[[X]]$ be a series with annihilating polynomial $P(y) = \sum_{n=0}^N c_n y^n$, where $c_0, \dots, c_N \in \mathbb{Z}[X]$ all have total degree at most D . Then the following hold:*

- 1) if $f \neq 0$ then $\text{ord}(f) \leq D$,
- 2) if $\deg(f) < \infty$ (f is a polynomial) then $\deg(f) \leq D$,
- 3) if $\deg(f) = \infty$ then $\text{ord}(\text{tail}_D(f)) \leq DN + D$.

Proof. For Item 1 we reason as follows. Polynomial P is non-zero by definition of an annihilating polynomial. Since f is assumed to be non-zero we can, by dividing P by a suitable power of y , assume without loss of generality that

the constant term c_0 is non-zero. From $P(f) = 0$ we have $c_0 = -\sum_{n=1}^N c_n f^n$ and thus

$$\begin{aligned} D &\geq \text{ord}(c_0) \\ &= \text{ord}\left(\sum_{n=1}^N c_n f^n\right) \\ &\geq \min\{\text{ord}(c_n f^n) : n = 1, \dots, N\} \\ &= \min\{\text{ord}(c_n) + n \cdot \text{ord}(f) : n = 1, \dots, N\} \\ &\geq \text{ord}(f). \end{aligned}$$

For Item 2, suppose that $\deg(f) < \infty$, i.e., f is a polynomial. Since $P(f) = 0$ we have $c_N f^N = -\sum_{n=0}^{N-1} c_n f^n$ and thus

$$\begin{aligned} N \deg(f) &\leq \deg(c_N f^N) \\ &= \deg\left(\sum_{n=0}^{N-1} c_n f^n\right) \\ &\leq \max\{\deg(c_n f^n) : n = 0, \dots, N-1\} \\ &\leq (N-1)\deg(f) + D. \end{aligned}$$

Hence $\deg(f) \leq D$.

For Item 3, suppose that $\deg(f) = \infty$. Then $g := \text{tail}_D(f)$ is a non-zero series with annihilating polynomial $Q(y) := P(y + f - g) = \sum_{n=0}^N \hat{c}_n y^n$. Since $\deg(f - g) \leq D$ we have that $\deg(\hat{c}_n) \leq ND + D$ for $n = 0, \dots, N$. Applying Item 1 we conclude that $\text{ord}(g) \leq ND + D$. \square

Combining Proposition 19 and Theorem 18 we obtain:

Corollary 20. *Let A be the strong solution of a proper polynomial system with ℓ variables and polynomials of degree at most d . Then for some absolute constant c and $D := d^{c\ell^2}$ we have:*

- ① if $A \neq 0$ then $\text{ord}(A) \leq D$,
- 2) if $\deg(A) < \infty$, then $\deg(A) \leq D$,
- 3) if $\deg(A) = \infty$, then $\text{ord}(\text{tail}_D(A)) \leq D^2 + D$.

Proof. By Theorem 18 there is a polynomial $P \in \mathbb{Z}[X][y]$ of total degree at most D such that $P(A) = 0$. Here c is the absolute constant mentioned in Theorem 18. Items 1–3 of the present result follow immediately from Items 1–3 of Proposition 19. \square

We will use Item 1 of Corollary 20 to decide EqAlg, and Items 2 and 3 for FinAlg.

D. Putting Things Together

The first main result of the section is as follows:

Theorem 2. *EqAlg is polynomial-time reducible to DegSLP and thereby lies in the counting class coRP^{PP} .*

Proof. Let S be a proper polynomial system of size s , over commuting indeterminates $X = (x_1, \dots, x_k)$, that has ℓ variables and involves polynomials of total degree at most d . Let A be the formal power series computed by S .

Write $D := d^{c\ell^2}$ for c the absolute constant mentioned in Corollary 20. By Equation (11), there is a polynomial

why not directly to EqSLP? because of trailing terms.

→ need to reverse the order when reducing to DegSLP

$\tilde{a} \in \mathbb{Z}[X]$ such that $\text{ord}(A - \tilde{a}) > D + 1$; and by Claim 14 there is an algorithm that constructs a circuit \mathcal{C} representing \tilde{a} in time $\text{poly}(s) + c\ell^2$.

It follows that $\text{ord}(\tilde{a}) \geq D + 1$ if A is identically zero and, by Corollary 20(1), $\text{ord}(\tilde{a}) \leq D$ if A is not identically zero.

Define $D' := 2^{s'}$ where s' is the size of the circuit \mathcal{C} . Note that D' is an upper bound of the degree of the polynomial represented by \mathcal{C} . One can construct a circuit \mathcal{C}' from \mathcal{C} , in time $\text{poly}(s)$, for the polynomial

$$f := (x_1 \cdots x_k)^{D'} \cdot \tilde{a} \left(\frac{1}{x_1}, \dots, \frac{1}{x_k} \right).$$

If A is identically zero then the total degree of f is at most $kD' - D - 1$, whereas if A is non-zero then the degree is at least $kD' - D$. \square

We now come to the second main result of the section, concerning the problem FinAlg . Before proceeding with the proof we observe that, unlike the approximating sequence of the strong solution, the sequence of iterates defined in Hensel's Lemma may fail to stabilise in finitely many steps even when the target sequence has finite support.

Example 3. Consider again the proper system $y = x + x^2 - 2xy + y^2$ discussed in Example 2. Although the unique quasiregular solution x has finite support, Hensel's iteration does not stabilise in finitely many steps, as witnessed by the closed-form formula for a_n .

Theorem 3. The complement of FinAlg is non-deterministic polynomial-time reducible to CoeffSLP , and thereby FinAlg lies in coNP^{PP} .

Proof. Let S be a proper polynomial system of size s , over commuting indeterminates $X = (x_1, \dots, x_k)$, that has ℓ variables and involves polynomials of total degree at most d . Let A be the formal power series computed by S .

Write $D := d^{c\ell^2}$ for c the absolute constant mentioned in Corollary 20. By Equation (11), there is a polynomial $\tilde{a} \in \mathbb{Z}[X]$ such that $\text{ord}(A - \tilde{a}) \geq D^2 + D$; and by Claim 14 there is an algorithm that constructs a circuit \mathcal{C} representing \tilde{a} in time $\text{poly}(s)$.

Suppose that A has finite support. Then $\deg(A) \leq D$ by Corollary 20(2) and hence \tilde{a} contains no monomial of total degree in the interval $[D + 1, D^2 + D]$. On the other hand, if A has infinite support then by Corollary 20(3) $\text{tail}_D(A)$ and hence also \tilde{a} contains some monomial of total degree $[D + 1, D^2 + D]$. We conclude that A has infinite support if and only if the polynomial \tilde{a} contains a monomial of total degree lying in the interval $[D + 1, D^2 + D]$. This monomial can be guessed and then checked for non-zerosness using an oracle for CoeffSLP . \square

V. APPLICATIONS TO CONTEXT-FREE GRAMMARS

The *Multiplicity Equivalence Problem* for context-free grammars asks, given two grammars G_1 and G_2 and respective nonterminals N_1, N_2 thereof, whether each word has the same number of derivations starting in N_1 as in N_2 (see below for

formal definitions). By taking the disjoint union of the two grammars, one may assume without loss of generality that N_1 and N_2 are non-terminals of the same grammar.

Decidability of multiplicity equivalence for grammars is a long-standing open problem in the theory of formal languages. It generalises decidability of language equivalence of unambiguous grammars, itself a recognised open problem, as well as decidability of language equivalence of deterministic pushdown automata, established in [25] (see also [26]). For the special case of linear context-free grammars with a distinguished symbol marking the middle of the word, multiplicity equivalence reduces to multiplicity equivalence of two-tape finite automata, which is known to be decidable for any number of tapes [27].

A. Context-Free Grammars

Let $\Sigma = \{\sigma_1, \dots, \sigma_k\}$ be a finite alphabet. The *Parikh image* is the function $c : \Sigma^* \rightarrow \mathbb{N}^k$ such that for all words w , we have $c(w) = (v_1, \dots, v_k)$ where the v_i is the number of occurrences of letter σ_i in w .

A *context-free grammar* is a tuple $G = (\Sigma, V, \Delta)$ where Σ is a finite alphabet, V is a set of *nonterminals*, and $\Delta \subseteq V \times (V \cup \Sigma)^+$ is a set of *production rules*. We write the production rules in the form $N \rightarrow \alpha$ where $N \in V$ and $\alpha \in (V \cup \Sigma)^+$. In this paper, we assume that the grammars are *proper*, that is the right-hand side α of each rule $N \rightarrow \alpha$ is non-empty and does not consist of a single non-terminal.

For the grammar G , the binary relation \Rightarrow on $(V \cup \Sigma)^*$, capturing a (*leftmost*) *derivation step*, is defined as follows: if $N \rightarrow \alpha$ is in Δ then for all words $\beta \in \Sigma^*$ and $\gamma \in (V \cup \Sigma)^*$ we have $\beta N \gamma \Rightarrow \beta \alpha \gamma$. A sequence $\alpha_0 \Rightarrow \alpha_1 \Rightarrow \dots \Rightarrow \alpha_k$ is a *derivation*, of α_k from α_0 .

The syntactic condition that G is proper implies that for all words $w \in \Sigma^*$, and all nonterminals $Y \in V$, the number of distinct derivations of w starting from Y is finite. For instance, for the improper grammar $X \rightarrow a, X \rightarrow X$ the word a has infinitely many derivations from X . We define the following *multiplicity semantics* $\llbracket N \rrbracket : \Sigma^* \rightarrow \mathbb{N}$ of a nonterminal N of a proper grammar G : For every finite word $w \in \Sigma^*$, $\llbracket N \rrbracket_w$ is the number of distinct derivations of w starting from the nonterminal N . The language of N , denoted $L(N)$, is the set of words w such that $\llbracket N \rrbracket_w \neq 0$.

The *Multiplicity Equivalence Problem* for a given grammar G and two nonterminals N_1, N_2 thereof asks whether $\llbracket N_1 \rrbracket = \llbracket N_2 \rrbracket$.

B. Letter-bounded languages

We say that a language $L \subseteq \Sigma^*$ is *letter-bounded* if there is an permutation $\sigma_1, \dots, \sigma_k$ of Σ such that $L \subseteq \sigma_1^* \cdots \sigma_k^*$. Deciding whether a given context-free language $L(G)$ is letter-bounded and moreover finding a witnessing enumeration of letters can be done in polynomial time by a simple dynamic programming algorithm: Then, we compute in polynomial time the set of letters that can appear as the first letter in some word of $L(G)$. If this set is not a singleton, then $L(G)$ is not letter-bounded. Otherwise, this set contains a

single letter σ_1 . Compute in polynomial time a grammar G_1 recognising the context-free language over $\Sigma_1 = \Sigma \setminus \{\sigma_1\}$ equal to $\{w \mid \exists n \in \mathbb{N} : \sigma_1^n w \in L(G), \sigma_1^{n+1} w \notin L(G)\}$. The procedure can be applied inductively to G_1 over the smaller alphabet Σ_1 . Since at each step we remove one letter from the alphabet, after k steps we reach a grammar G_k over the empty alphabet $\Sigma_k = \emptyset$. Then $L(G)$ is letter-bounded iff $L(G_k) = \emptyset$; in the positive case, the algorithm has constructed a witnessing enumeration $\sigma_1^* \cdots \sigma_k^*$. Moreover, each of the k steps is performed in polynomial time, so we have an overall polynomial time complexity.

We show how to decide multiplicity equivalence of non-terminals generating a letter-bounded language. Consider a nonterminal N . Recall that $\llbracket N \rrbracket_w \in \mathbb{N}$, for a word $w \in \Sigma^*$, is the multiplicity of w as generated by N . We now aggregate the multiplicities of all words with the same Parikh image. Let $X = (x_1, \dots, x_k)$ be a tuple of commuting indeterminates, with one variable x_i for each terminal symbol σ_i . Given a non-terminal N , define its *census* generating function [7] to be the multivariate power series

$$f_N := \sum_{\mathbf{v} \in \mathbb{N}^k} a_{\mathbf{v}}(N) \cdot X^{\mathbf{v}},$$

where

$$a_{\mathbf{v}}(N) := \sum_{c(w)=\mathbf{v}} \llbracket N \rrbracket_w.$$

Thus defined, the tuple of formal series $(f_N)_{N \in V}$ satisfies a proper polynomial system over indeterminates X that can be obtained from the grammar G in polynomial time [1, Theorem 1.5 in Chapter IV]. This system is obtained by a classic syntactic transformation applied to the grammar. Rather than formally defining it here, we present it with an example.

Example 4. Consider the proper grammar G over the alphabet of terminal symbols $\Sigma = \{a, b, c, d\}$ with nonterminal symbols $V = \{X, Y\}$ and a production rules

$$X \rightarrow ab, X \rightarrow aXXb, X \rightarrow cYd, Y \rightarrow cd, Y \rightarrow cYYd.$$

We obtain the proper system of polynomial equations:

$$f_X = x_1x_2 + x_1x_2f_X^2 + x_3x_4f_Y \quad f_Y = x_3x_4 + x_3x_4f_Y^2.$$

Multiplicity equivalence of letter-bounded context-free grammars reduces to equivalence of census generating functions.

Lemma 21. Let N_1, N_2 be non-terminals of a grammar such that $L(N_1), L(N_2) \subseteq \sigma_1^* \cdots \sigma_k^*$. Then

$$\llbracket N_1 \rrbracket = \llbracket N_2 \rrbracket \quad \text{if and only if} \quad f_{N_1} = f_{N_2}.$$

Proof. This follows at once from the fact that the Parikh image c restricted to $\sigma_1^* \cdots \sigma_k^*$ is a bijection onto \mathbb{N}^k . As a consequence, for every non-terminal N and every vector $\mathbf{v} = (v_1, \dots, v_k) \in \mathbb{N}^k$ we have

$$\llbracket N \rrbracket_{\sigma_1^{v_1} \cdots \sigma_k^{v_k}} = a_{\mathbf{v}}(N).$$

□

Lemma 21 shows that multiplicity equivalence of letter-bounded context-free grammars is a special case of EqAlg for the census generating function. By Theorem 2 the latter can be decided in **coRP^{PP}**, thus proving Corollary 5(1) from the introduction.

C. Bounded context-free languages

A language $L \subseteq \Sigma^*$ is *bounded* if there exist nonempty words $w_1, \dots, w_k \in \Sigma^+$ such that $L \subseteq w_1^* \cdots w_k^*$. Many algorithmic problems on context-free grammars are more tractable on bounded languages. Checking whether a given context-free grammar recognises a bounded language is decidable [28, Theorem 5.5.2], and can be done in polynomial time [29, Theorem 19]. We note however that there are grammars recognising a bounded language where the number of witnessing words k is exponential in the size of the grammar.

In this section we give complexity bounds for deciding multiplicity equivalence for arbitrary grammars *restricted* to a bounded language $L := w_1^* \cdots w_k^*$ that is explicitly given by the list of words w_1, \dots, w_k . This problem asks to decide whether $\llbracket N_1 \rrbracket_w = \llbracket N_2 \rrbracket_w$ for all words $w \in L$. We reduce the restricted multiplicity equivalence problem to the letter-bounded case.

Lemma 22. The Multiplicity Equivalence Problem restricted to a bounded language reduces in polynomial time to the multiplicity equivalence problem for grammars recognising a letter-bounded language.

Proof. Suppose we wish to check multiplicity equivalence of two non-terminals of a grammar G , restricted to a bounded language $L := w_1^* \cdots w_k^*$. Consider a fresh alphabet $\Gamma = \{a_1, \dots, a_k\}$ and define the homomorphism $h : \Gamma^* \rightarrow \Sigma^*$ by setting $h(a_1) = w_1, \dots, h(a_k) = w_k$. Below, we combine classical constructions to transform G into a new grammar G' with the property that for each non-terminal N of G there is a non-terminal N' of G' such that

$$\llbracket N' \rrbracket_w = \llbracket N \rrbracket_{h(w)}. \quad (15)$$

This transformation is done in three steps. In the first step, as in [2, Theorem 14.33], we convert in polynomial time the non-terminal N to a pushdown automaton A with the same multiplicity semantics: $\llbracket N \rrbracket = \llbracket A \rrbracket$ where $\llbracket A \rrbracket$ is the function that maps each word w to the number of accepting runs of A over w .

In the second step, from A we build a pushdown automaton B recognising the inverse homomorphic image of the language recognised by A : $L(B) = h^{-1}(L(A))$. This can be achieved by a standard construction on pushdown automata [30, Theorem 7.30], which can be performed in polynomial time. Inspecting the construction, we observe that that multiplicities are preserved, meaning that, for every word $w \in \Gamma^*$, the equality $\llbracket A \rrbracket_{h(w)} = \llbracket B \rrbracket_w$ holds (see also [31, Claim 11]).

In the third step, as in [2, Theorem 14.15], we convert in polynomial time the pushdown automaton B to a nonterminal N' having the same multiplicity semantics: $\llbracket B \rrbracket = \llbracket N' \rrbracket$. This establishes (15).

The non-terminal N' defined above need not recognise a letter-bounded language. This can be remedied by taking the product of N' with a deterministic finite automaton A for the language $a_1^* \cdots a_k^*$, which is multiplicity preserving and can be done in polynomial time (as described in [1, Chapter IV, Theorem 3.5]). Let N'_1 and N'_2 be the nonterminals obtained from the product of N'_1 , resp., N'_2 with the automaton A . Not only are N'_1 and N'_2 letter bounded, but $\llbracket N'_1 \rrbracket$ and $\llbracket N'_2 \rrbracket$ coincide on L just in case $\llbracket N'_1 \rrbracket = \llbracket N'_2 \rrbracket$. \square

Combining Corollary 5(1) and Lemma 22, we obtain Corollary 5(2) from the introduction.

VI. DISCUSSION

In this paper we have investigated a number of computational problems concerning the coefficients of algebraic series defined by systems of polynomial equations and have related these to analogous well-known problems for arithmetic circuits. In Theorem 1 we showed that the problems CoeffAlg and CoeffSLP are polynomial-time interreducible. A natural question for future work is whether it is likewise possible to reduce EqAlg to EqSLP , rendering these two problems equivalent under polynomial-time reductions. In Theorem 2 we gave a reduction of EqAlg to DegSLP . It is easily seen that EqSLP reduces in polynomial time to DegSLP , but it is not known whether there is a polynomial-time reduction in the other direction. There is moreover a significant difference in the best known complexity upper bounds for the two problems: EqSLP (i.e., polynomial identity testing) famously admits a number of different randomised polynomial-time algorithms, whereas the best known complexity bound for DegSLP involves a randomised polynomial-time algorithm with a **PP** oracle.

ACKNOWLEDGMENT

We would like to thank Wojciech Czerwiński for pointing out reference [29] to us. We would like to thank Keith Conrad for his invaluable collection of expository papers, in particular his note on the multivariate version of Hensel's lemma.

REFERENCES

- [1] A. Salomaa and M. Soittola, *Automata-theoretic Aspects of Formal Power Series*, ser. Texts and monographs in computer science. Springer-Verlag, 1978.
- [2] W. Kuich and A. Salomaa, *Semirings, Automata, Languages*, ser. Monographs in Theoretical Computer Science. An EATCS Series. Springer Berlin Heidelberg, 1986.
- [3] E. Allender, P. Bürgisser, J. Kjeldgaard-Pedersen, and P. B. Miltersen, "On the complexity of numerical analysis," *SIAM J. Comput.*, vol. 38, no. 5, pp. 1987–2006, Jan. 2009.
- [4] N. Kayal and C. Saha, "On the sum of square roots of polynomials and related problems," *ACM Transactions on Computation Theory (TOCT)*, vol. 4, no. 4, pp. 1–15, 2012.
- [5] V. Forejt, P. Jancar, S. Kiefer, and J. Worrell, "Language equivalence of probabilistic pushdown automata," *Inf. Comput.*, vol. 237, pp. 1–11, 2014. [Online]. Available: <https://doi.org/10.1016/j.ic.2014.04.003>
- [6] B. E. Litow, "Computing a context-free grammar-generating series," *Inf. Comput.*, vol. 169, no. 2, pp. 174–185, 2001.
- [7] A. Panholzer, "Gröbner bases and the defining polynomial of a context-free grammar generating function," *J. Autom. Lang. Comb.*, vol. 10, no. 1, pp. 79–97, 2005. [Online]. Available: <https://doi.org/10.25596/jalc-2005-079>

- [8] L. Clemente, "On the complexity of the universality and inclusion problems for unambiguous context-free grammars," in *Proceedings 8th International Workshop on Verification and Program Transformation and 7th Workshop on Horn Clauses for Verification and Synthesis*, Dublin, Ireland, 25–26th April 2020, ser. EPTCS, L. Fribourg and M. Heizmann, Eds., vol. 320. Open Publishing Association, 2020, pp. 29–43.
- [9] J. Esparza, S. Kiefer, and M. Lüttenberger, "Newtonian program analysis," *J. ACM*, vol. 57, no. 6, pp. 33:1–33:47, 2010.
- [10] K. Etessami, D. Wojtczak, and M. Yannakakis, "Quasi-birth-death processes, tree-like qbds, probabilistic 1-counter automata, and pushdown systems," *Perform. Eval.*, vol. 67, no. 9, pp. 837–857, Sep. 2010.
- [11] K. Etessami, A. Stewart, and M. Yannakakis, "Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars," in *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, ser. STOC '12. New York, NY, USA: Association for Computing Machinery, 2012, pp. 579–588.
- [12] K. Etessami and M. Yannakakis, "Recursive markov chains, stochastic grammars, and monotone systems of nonlinear equations," *J. ACM*, vol. 56, no. 1, pp. 1:1–1:66, Feb. 2009.
- [13] K. Etessami, A. Stewart, and M. Yannakakis, "Stochastic context-free grammars, regular languages, and newton's method," in *Proc. of ICALP '13*, ser. ICALP '13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 199–211.
- [14] C. Banderier and M. Drmota, "Formulae and asymptotics for coefficients of algebraic functions," *Comb. Probab. Comput.*, vol. 24, no. 1, pp. 1–53, 2015.
- [15] S. Arora and B. Barak, *Computational Complexity: A Modern Approach*. Cambridge University Press, 2006.
- [16] S. J. Berkowitz, "On computing the determinant in small parallel time using a small number of processors," *Information processing letters*, vol. 18, no. 3, pp. 147–150, 1984.
- [17] M. Mahajan and V. Vinay, "Determinant: Old algorithms, new insights," *SIAM journal on Discrete Mathematics*, vol. 12, no. 4, pp. 474–490, 1999.
- [18] P. Koiran and S. Perifel, "The complexity of two problems on arithmetic circuits," *Theoretical Computer Science*, vol. 389, no. 1, pp. 172–181, 2007. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0304397507006561>
- [19] N. Bourbaki, *Commutative Algebra: Chapters 1–7*, ser. Elements de mathématique. English. Springer, 1998, no. vol. 1.
- [20] D. Eisenbud, *Commutative algebra: with a view toward algebraic geometry*. New York: Springer Verlag, 1995.
- [21] "Lectures in the model theory of valued fields," in *Model Theory in Algebra, Analysis, and Arithmetic*, ser. Lecture Notes in Mathematics. Heidelberg: Springer, 2014, vol. 2111, pp. 55–157.
- [22] J. Berstel and C. Reutenauer, *Noncommutative Rational Series with Applications*. Cambridge University Press, 2010.
- [23] V. Strassen, "Vermeidung von divisionen," *Journal für die reine und angewandte Mathematik*, vol. 264, pp. 184–202, 1973.
- [24] S. Basu, R. Pollack, and M. Roy, *Algorithms in Real Algebraic Geometry*, ser. Algorithms and Computation in Mathematics. Springer Berlin Heidelberg, 2006.
- [25] G. Sénizergues, "The equivalence problem for deterministic pushdown automata is decidable," in *Proc. of ICALP '97*, P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 671–681.
- [26] P. Jancar, "Decidability of dpda language equivalence via first-order grammars," in *27th Annual IEEE Symposium on Logic in Computer Science*, 2012, pp. 415–424.
- [27] T. Harju and J. Karhumäki, "The equivalence problem of multitape finite automata," *Theoretical Computer Science*, vol. 78, no. 2, pp. 347–355, 1991.
- [28] S. Ginsburg, *The mathematical theory of context-free languages*. McGraw-Hill, Inc., 1966.
- [29] P. Gawrychowski, D. Krieger, N. Rampersad, and J. Shallit, "Finding the growth rate of a regular or context-free language in polynomial time," *International Journal of Foundations of Computer Science*, vol. 21, no. 04, pp. 597–618, 2010.
- [30] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Addison-Wesley, 2000.
- [31] D. Raz, "Deciding Multiplicity Equivalence for Certain Context-free Languages," in *Proceedings of the First International Conference on Developments in Language Theory*. World Scientific, 1993, pp. 18–29.

APPENDIX

A. Extended Preliminaries

Proposition 6. *Given $m \in \mathbb{N}$, there is a circuit of size $O(\log m)$ that represents the polynomial $\sum_{i=0}^m x^i$.*

Proof. Define $S_m := \sum_{i=0}^m x^i$. Note that

$$\begin{bmatrix} S_{m+1} \\ 1 \end{bmatrix} = \begin{bmatrix} x & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} S_m \\ 1 \end{bmatrix}$$

and thus

$$\begin{bmatrix} S_m \\ 1 \end{bmatrix} = \begin{bmatrix} x & 1 \\ 0 & 1 \end{bmatrix}^m \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

for all m . Since exponentiation of a matrix to the power m can be implemented via $O(\log m)$ steps of repeated squaring, the statement follows. \square

Claim 9. *For a univariate polynomial $p(x) \in R[x]$ and two elements $a, b \in R$, $v(p(a) - p(b)) \geq v(a - b)$.*

Proof. Let $p(x) = c_0x^0 + \dots + c_nx^n$. We then have $p(a) - p(b) = (a - b) \cdot q(a, b)$ for some polynomial $q(x, y) \in R[x, y]$. Thus $v(p(a) - p(b)) = v(a - b) + v(q(a, b)) \geq v(a - b)$, where the last inequality follows from the fact that the valuation is nonnegative. \square

B. Missing proofs in Sec. III

Claim 11. $J_{\mathbf{f}}(\mathbf{a}) \in 1 + \mathfrak{m}$ for all $\mathbf{a} \in \mathfrak{m}^\ell$.

Proof. Recall that the derivative matrix of \mathbf{f} is

$$D\mathbf{f} = \begin{bmatrix} \frac{\partial f_1}{\partial y_1} & \dots & \frac{\partial f_\ell}{\partial y_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_1}{\partial y_\ell} & \dots & \frac{\partial f_\ell}{\partial y_\ell} \end{bmatrix} \quad (16)$$

Recall also that S is assumed to be a proper equation system. This requires that for each polynomial P_i and all monomial $Y^{\mathbf{v}}$ of total degree at most 1 that appears in P_i , the coefficient $a_{\mathbf{v}} \in \mathbb{Z}[X]$ lies in \mathfrak{m} . Consequently, we can write f_i as

$$f_i = y_i - h_{i,0} - \sum_{j=1}^{\ell} h_{i,j} y_j - \sum_{|\mathbf{v}| > 1} g_{i,\mathbf{v}} Y^{\mathbf{v}}$$

for some $g_{i,\mathbf{v}} \in R_0$ and $h_{i,j} \in \mathfrak{m}$ for all $j \in \{0, \dots, \ell\}$.

Let $i, j \in \{1, \dots, \ell\}$ be such that $i \neq j$. We can write $\frac{\partial f_i}{\partial y_i} \in R_0[Y]$ as

$$\frac{\partial f_i}{\partial y_i} = 1 - h_{i,i} - \sum_{\substack{|\mathbf{v}| > 1 \\ v_i \geq 1}} v_i g_{i,\mathbf{v}} Y^{\mathbf{v} - e_i}$$

where e_i is the i -th vector in the standard basis (with a 1 in the i -th coordinate and 0 elsewhere). Analogously, we can write $\frac{\partial f_i}{\partial y_j} \in R_0[Y]$ as

$$\frac{\partial f_i}{\partial y_j} = -h_{i,j} - \sum_{\substack{|\mathbf{v}| > 1 \\ v_j \geq 1}} v_j g_{i,\mathbf{v}} Y^{\mathbf{v} - e_j}$$

Consequently, for all $\mathbf{a} \in \mathfrak{m}^\ell$, we have

$$\frac{\partial f_i}{\partial y_i}(\mathbf{a}) \in 1 + \mathfrak{m}$$

and

$$\frac{\partial f_i}{\partial y_j}(\mathbf{a}) \in \mathfrak{m}.$$

Thus $D\mathbf{f}$ is entry-wise congruent to the identity matrix modulo \mathfrak{m} . Since the determinant of a matrix is a polynomial in its entries, it follows that $J_{\mathbf{f}}(\mathbf{a})$ is congruent modulo \mathfrak{m} to the determinant of the identity matrix, which proves the claim. \square

Claim 12. *For all $\mathbf{a} \in \mathfrak{m}^\ell$ we have $\mathbf{f}(\mathbf{a}) \in \mathfrak{m}^\ell$.*

Proof. Recall that \mathcal{S} is assumed to be a proper equation system. This requires that for each polynomial P_i and all monomials $Y^\mathbf{v}$ of total degree at most 1 that appear in P_i , the coefficient $a_\mathbf{v} \in \mathbb{Z}[X]$ lies in \mathfrak{m} . Suppose that $\mathbf{a} \in \mathfrak{m}$. Then

$$f_i(\mathbf{a}) = y_i - P_i(\mathbf{a})$$

But $P_i(\mathbf{a}) \in \mathfrak{m}$ since the coefficient of Y^0 lies in \mathfrak{m} . □

Claim 13. *There is an algorithm that, given an equation system \mathcal{S} , $i \in \{1, \dots, \ell\}$, and $n \in \mathbb{N}$, produces circuits $C_{n,i}, D_{n,i}$ representing polynomials $g_{n,i}, h_{n,i} \in R_0$ respectively, as in Equation (8). The algorithm runs in $\mathbf{poly}(s, n)$ time and hence $C_{n,i}, D_{n,i}$ have size $\mathbf{poly}(s, n)$.*

We recall here Equation (8) mentioned in the statement of the claim:

$$\mathbf{a}_{n,i} = \frac{g_{n,i}}{1 - h_{n,i}}, \quad \text{with } \text{ord}(h_{n,i}) \geq 1. \quad (8)$$

Proof of Claim 13. Recall that $\mathbf{a}_0 = \mathbf{0}$ and that we have the recursive formula (6). We define the circuits $C_{n,i}$ and $D_{n,i}$ by induction on n . For each $i \in \{1, \dots, \ell\}$ we define $C_{0,i}$ to be the constant 0 and $D_{0,i}$ to be the constant 1. For the inductive step we show how to construct for each $i \in \{1, \dots, \ell\}$ the circuits $C_{n+1,i}$ and $D_{n+1,i}$ from the collection of circuits $\{C_{n,j}, D_{n,j} : 1 \leq j \leq \ell\}$, in time $\mathbf{poly}(s)$. For this we will use Equation (6).

Proposition 7 ensures we can construct $\mathbf{poly}(s)$ -size circuits for $J_{\mathbf{f}}(\mathbf{a}_n)$ as well as every entry of adjugate $\text{Adj}(D\mathbf{f}(\mathbf{a}_n))$, whose entries are just cofactors of $D\mathbf{f}(\mathbf{a}_n)$. Composing the circuits so-obtained with the circuits $\{C_{n,j}, D_{n,j} : 1 \leq j \leq \ell\}$ that represent the respective numerators and denominators of the entries of \mathbf{a}_n we obtain the desired circuits $C_{n+1,i}$ and $D_{n+1,i}$. This can be easily done by using the rules $\frac{A}{B} + \frac{C}{D} = \frac{AD+BC}{BD}$ and $\frac{A}{B} \cdot \frac{C}{D} = \frac{AC}{BD}$.

It remains to argue that $\text{ord}(h_{n,i}) \geq 1$ for the polynomial $h_{n,i} \in R_0$ represented by $D_{n,i}$, where $\mathbf{a}_{n,i} = \frac{g_{n,i}}{1-h_{n,i}}$. Since $J_{\mathbf{f}}(\cdot)$ is a polynomial map, it maps $(\mathbb{Q}(X) \cap R)^\ell$ to the ring $\mathbb{Q}(X) \cap R$. In particular, we can write $J_{\mathbf{f}}(\mathbf{a}_{n-1})$ as $\frac{s}{1-t}$ for some $s, t \in R_0$. Moreover, by Claim 11 we know that $J_{\mathbf{f}}(\mathbf{a}_{n-1}) \in 1 + \mathfrak{m}$, hence the constant term of s must be 1. Since $h_{n,i}$ is obtained through multiplication of s and some of the polynomials $1 - h_{n-1,j}$ with $j \in \{1, \dots, \ell\}$ and $\text{ord}(h_{n-1,j}) \geq 1$, the denominator of $\mathbf{a}_{n,i}$ will have constant term 1. The claim follows. □

Claim 14. *There is an algorithm that, given an equation system \mathcal{S} and $n \in \mathbb{N}$, produces circuit E_n representing the polynomial $\tilde{\mathbf{a}}_{n,1}$, defined in Equation (12) for $i = 1$. The algorithm runs in $\mathbf{poly}(s, n)$ time, and E_n has size $\mathbf{poly}(s, n)$.*

Proof. By definition in Equation (12), we have $\tilde{\mathbf{a}}_{n,1} = g_{n,1} \sum_{j=0}^{2^n-1} h_{n,1}^j$. Note that a naive circuit for the expression on the right-hand side would have size $\mathbf{poly}(s)2^n$. However one can build a circuit of size $\mathbf{poly}(s, n)$ by plugging in $x = h_{n,1}$ and $m = 2^n$ in the circuit from Proposition 6. □

Claim 15. *There is a polynomial-time reduction from CoeffSLP to CoeffAlg.*

Proof. We present a straightforward reduction from CoeffSLP to CoeffAlg. Fix an instance of CoeffSLP, comprising a polynomial f represented by an arithmetic circuit C over variables X , a monomial $X^\mathbf{v}$ and a prime p .

We say that C is balanced if all paths from the input gates to the designated output gate have equal length. It is folklore that an algebraic circuit of size s can be transformed into a binary and balanced circuit of size $O(s)$ computing the same polynomial in time $O(s)$. Furthermore, the equivalent circuit can be constructed to have alternating levels of multiplication and addition/subtraction gates. We thus assume without loss of generality that the circuit C is binary, balanced and alternating.

Fix $\tilde{x} \in X$. We first replace each input gate $m \in \{0, 1\} \cup X$ in C with $m\tilde{x}$. Next, we consider the corresponding SLP of C . That is a collection of sequential instructions in the form

$$Y_i := Y_j \odot Y_k$$

with $\odot \in \{\times, +, -\}$ where the Y_i is an \odot -gate, and the Y_j, Y_k are the two inputs of Y_i in C .

We modify these equations to obtain a proper equation system \mathcal{S} . The idea is to merge three equations arising from a multiplication followed by additions/subtractions into a single equation. The construction merges equations such as

$$\begin{aligned} Y_1 &:= Y_2 \times Y_3 \\ Y_2 &:= Y_4 + Y_5 \\ Y_3 &:= Y_6 + Y_7 \end{aligned}$$

with all the Y_i variables in the SLP, to the single equation

$$Y_1 := Y_4 Y_6 + Y_4 Y_7 + Y_5 Y_6 + Y_5 Y_7.$$

A similar transformation is applied to a multiplication gate whose inputs are two subtraction gates, or an addition and a subtraction gate.

We argue the power series computed by \mathcal{S} , that is clearly proper, is f multiplied with \tilde{x}^α for some $\alpha \geq 1$. The proof follows from the assumption that C is balanced. The proof is by an induction showing that all gates with the same distance to the input gates are multiplied with \tilde{x}^α for some α . For the base of induction, observe that all input gates m are replaced with $m\tilde{x}$. For the induction step, given that two inputs Y_j and Y_m of a \odot -gate Y_i are both multiplied with x^α , then Y_i is multiplied with

- \tilde{x}^α if $\odot \in \{+, -\}$,
- $\tilde{x}^{2\alpha}$ otherwise (i.e., $\odot = \times$).

We note that one can compute in NC an α such that the power series computed by \mathcal{S} is precisely $\tilde{x}^\alpha f$. The statement of the claim follows. \square

C. Proof of Equation (10)

We will prove by induction that, for all $n \geq 1$, the closed-form formula (10) for a_n holds. It is straightforward to verify that $a_1 = x - \frac{x^2}{(x+1)^2 - x^2}$. For the inductive step, assume that (10) holds for some $n \in \mathbb{N}$. Following (9) we can write

$$\begin{aligned}
 a_{n+1} &= \frac{-(x - \frac{x^{2^n}}{(x+1)^{2^n} - x^{2^n}})^2 + x^2 + x}{1 + 2x - 2(x - \frac{x^{2^n}}{(x+1)^{2^n} - x^{2^n}})} \\
 &= \frac{2x \frac{x^{2^n}}{(x+1)^{2^n} - x^{2^n}} - \frac{(x^{2^n})^2}{((x+1)^{2^n} - x^{2^n})^2} + x}{1 + \frac{x^{2^n}}{(x+1)^{2^n} - x^{2^n}}} \\
 &= \frac{2x \cdot x^{2^n} (x+1)^{2^n} - 2x \cdot x^{2^{(n+1)}} - x^{2^{(n+1)}} + x \cdot ((x+1)^{2^n} - x^{2^n})^2}{((x+1)^{2^n} - x^{2^n} + 2x^{2^n})((x+1)^{2^n} - x^{2^n})} \tag{17} \\
 &= \frac{x \cdot (x+1)^{(n+1)} - x \cdot x^{2^{(n+1)}} - x^{2^{(n+1)}}}{((x+1)^{2^n} + x^{2^n})((x+1)^{2^n} - x^{2^n})} \\
 &= \frac{x \cdot (x+1)^{(n+1)} - x \cdot x^{2^{(n+1)}} - x^{2^{(n+1)}}}{(x+1)^{2^{(n+1)}} - x^{2^{(n+1)}}}
 \end{aligned}$$

Hence $a_{n+1} = x - \frac{x^{2^{(n+1)}}}{(x+1)^{2^{(n+1)}} - x^{2^{(n+1)}}}$ and our induction is completed.