When Polynomial Equation Systems Can Be "Solved" Fast? ⋆

M. Giusti [1], J. Heintz [2], J.E. Morais [2] and L.M. Pardo [2]

[1] GAGE, Centre de Mathématiques, École Polytechnique, F-91128, Palaiseau Cedex, FRANCE
[2] Departamento de Matemáticas, Estadística y Computación, Facultad de Ciencias, Universidad de Cantabria, E-39071 SANTANDER, Spain

Abstract. We present a new method for solving symbolically zero–dimensional polynomial equation systems in the affine and toric case. The main feature of our method is the use of an alternative data structure : arithmetic networks and straight–line programs with FOR gates. For sequential time complexity measured by the size of these networks we obtain the following result : it is possible to solve any affine or toric zero–dimensional equation system in non–uniform sequential time which is polynomial in the length of the input description and the "geometric degree" of the equation system. Here, the input is thought to be given by a straight–line program (or alternatively in sparse representation), and the length of the input is measured by number of variables, degree of equations and size of the program (or sparsity of the equations). Geometric degree has to be adequately defined. It is always bounded by the algebraic–combinatoric "Bézout number" of the system which is given by the Hilbert function of a suitable homogeneous ideal. However, in many important cases, the value of the geometric degree is much smaller than the Bézout number since it does not take into account multiplicities or degrees of extraneous components (which are at infinity in the affine case or contained in some coordinate hyperplane in the toric case).
Finally, we announce the result that FOR gates can be avoided by a method which, based on Newton iteration, pulls back the whole question to ordinary arithmetic networks and straight–line programs. In this context, our complexity bounds remain valid. However, this second procedure is not rational anymore because it requires computing with algebraic numbers. This is due to its numeric ingredients (Newton iteration). Nevertheless, at least in the case of polynomial equation systems depending on parameters, the practical advantage of our method with respect to more traditional ones in symbolic and numeric computation is clearly visible.
It should be well understood that our method does not improve the well known worst–case complexity bounds for zero–dimensional equation solving in symbolic and numeric computing.

⋆ Research was partially supported by the following european, french and spanish grants : PoSSo ESPRIT/BRA 6846, GDR CNRS 1026 MEDICIS, DGICT PB 92–0498–C02–01 and PB93–0472–C02–02

Keywords. Polynomial system solving, elimination, geometric degree, arithmetic networks, straight–line programs, complexity.

1 Introduction and Statement of Results

Let k be an infinite, perfect field which we think "effective" with respect to arithmetic operations as addition/subtraction, multiplication/division and extraction of p–th roots in the case of positive characteristic p. Let $\bar{k}$ be a fixed algebraic closure of k.

An important problem in elimination theory is the computation of the isolated points of an affine algebraic variety. This means we consider the algebraic variety

$$V := V(f_1, \ldots, f_s) := \{x \in \mathbf{A}^n(\bar{k}) : \, f_1(x) = 0, \ldots, f_s(x) = 0\},$$

where $f_1, \ldots, f_s \in k[X_1, \ldots, X_n]$ with $s \geq n$ are polynomials of degree at most d in the variables $X_1, \ldots, X_n$ and we look at the following problem : given a non–zero linear form $H \in k[X_1, \ldots, X_n]$, compute a non–zero polynomial $p \in k[T]$ such that $p(H)$ vanishes on the isolated points of V.

As many authors do (cf. for instance, [2], [3], [6], [7], [8], [9], [11], [15], [17], [21], [22], [33], [34], [35], [36], [42], [43]), we replace the original input system by n generic linear combinations of the equations $f_1, \ldots, f_s$. This preparation of the input conserves all irreducible components (and, in particular, the isolated points) of V, adding possibly some new, extraneous ones (these components have to be eliminated afterwards). Thus, let us suppose from now on that $s = n$ holds. This implies that the input system $f_1, \ldots, f_n$ forms locally a complete intersection with respect to the isolated points of V. In view of the surprisingly low (linear) regularity bound for homogeneous global complete intersection ideals (cf. [5], [37], [38]) one is tempted to replace the original equations $f_1, \ldots, f_n \in k[X_1, \ldots, X_n]$ by new homogeneous ones, $G_1, \ldots, G_n \in k[X_0, \ldots, X_n]$, and the affine variety $V = V(f_1, \ldots, f_n)$ by the corresponding projective one, $W := V(G_1, \ldots, G_n) = \{x \in \mathbb{P}^n(\bar{k}) \; ; \; G_1(x) = 0, \ldots, G_n(x) = 0\}$. A simple minded way to proceed consists in taking as $G_1, \ldots, G_n$ just the homogenizations of the original system $f_1, \ldots, f_n$. However this idea does not produce the desired effect even in case that the polynomials $f_1, \ldots, f_n$ form a regular sequence in $k[X_1, \ldots, X_n]$. The reason is that the homogenizations of $f_1, \ldots, f_n$ need not to generate a complete intersection ideal.

In order to remedy this defect, many authors use homotopic deformations and path–following methods for the construction of a regular sequence of auxiliar homogeneous polynomials $G_1, \ldots, G_n$, from whose zeroes the solutions of the original system $f_1, \ldots, f_n$ are then extracted (see [1], [9], [10], [11], [14], [19], [21], [23], [34], [35], [36], [39], [45]). The polynomials $G_1, \ldots, G_n$ are homogeneous in the variables $X_0, X_1, \ldots, X_n$. Moreover they depend on a deformation parameter ε and form a regular sequence in $k(\varepsilon)[X_0, X_1, \ldots, X_n]$. They define a zero dimensional projective subvariety of $\mathbb{P}^n(\overline{k(\varepsilon)})$ without points at infinity (here $\overline{k(\varepsilon)}$ denotes any algebraic closure of the function field $k(\varepsilon)$). Specializing the polynomials $G_1, \ldots, G_n$ in $\varepsilon = 0$ we obtain a projective subvariety of $\mathbb{P}^n(\bar{k})$ which contains the isolated points of V as irreducible components. Since the underlying deformation is flat in the isolated points of V well known techniques based on the implicit or explicit use of Macaulay's u–resultant allow first to find the solutions of $V(G_1, \ldots, G_n)$ and finally the isolated points of V.

This deformation method introduces a somehow artificial dependency of the complexity of the algorithms on the regularity of the Hilbert function of the homogeneous ideal $(G_1, \ldots, G_n)$ of $k(\varepsilon)[X_0, X_1, \ldots, X_n]$ and on the degree of $(G_1, \ldots, G_n)$ which is defined by means of the Hilbert polynomial. The regularity of $(G_1, \ldots, G_n)$ is bounded by $nd - n$. This implies that deformation based algorithms have to triangulate rectangular matrices of size $\binom{nd}{n} \times n\binom{nd}{n}$ in order to extract from this data some "geometrically meaningful" square matrix of size the Bézout number. The characteristic polynomial of this square matrix is the basic eliminating form that all algorithms work with. However, the Bézout number of the ideal $(G_1, \ldots, G_n)$ (i.e. the degree of the eliminating form) includes multiplicities and degrees of extraneous components which the previous deformation process adds to the original variety V. The exact value of the Bézout number is $\prod_{i=1}^{n} deg G_i$, roughly d^n.

The method we present in this paper will be independent of "algebraic" quantities such as regularity of the Hilbert function or Bézout number of an appropriate homogeneous ideal. This allows in geometrically well suited cases (when the geometric degree of the variety V is low) to reduce the size of the matrices in the algorithms (and hence the complexity of the procedures). However, in worst case (e.g. when the equations $f_1, \ldots, f_n$ are generic) our complexity bounds are roughly the same as those obtained by deformation based algorithms. This is not surprising in view of the known lower bounds for elimination problems ([27], [46]).

An important aspect concerning elimination procedures consists in the encoding of the polynomials appearing as inputs, outputs or intermediate results of our algorithms. Encoding polynomials by their coefficients (dense representation) faces us with an input size of order $\binom{d+n}{n} \leq c \cdot d^n$ (if $n, d \geq 3$ we can take $c = 1$, otherwise we can take as c the base of the natural logarithm). If the equations $f_1, \ldots, f_n$ are generic, the geometric degree of the variety V and the Bézout number of the system coincide and are of order d^n. Thus, a complexity bound of type $d^{O(n)}$ is both polynomial in the dense input size and the degree of the variety V, what is the best we can hope in worst case (see [27]).

However, when the geometric degree of V is low, one wishes to use more economic encodings of inputs, outputs and intermediate results. Such an encoding is given, for instance, by the data structure straight–line program.

From now on let the following notations and assumptions be fixed :

There is given a family of n input polynomials $f_1, \ldots, f_n \in k[X_1, \ldots, X_n]$ which are thought to be encoded by a straight–line program (arithmetic circuit)

β without essential divisions in $k[X_1, \ldots, X_n]$ (this means we allow β only to contain divisions by non–zero elements of k). We denote the nonscalar size ("nonscalar length" in more traditional terminology) and the nonscalar depth of β by L and ℓ respectively (see [20], [26], [34], [47] and Section 2 for the notions of straight–line program, arithmetic network and complexity measures we shall use subsequently).

Let us first consider the *affine case*. Here, we suppose that $f_1, \ldots, f_n$ form a regular sequence in $k[X_1, \ldots, X_n]$. Fix $1 \le i \le n$. The affine variety defined by the ideal $(f_1, \ldots, f_i)$, namely $V(f_1, \ldots, f_i)$, is a Zariski closed subset of $\mathbf{A}^n := \mathbf{A}^n(\bar{k})$ of *pure dimension* $n-i$ (i.e. all irreducible components of $V(f_1, \ldots, f_i)$ have the same dimension $n-i$; we say also that $V(f_1, \ldots, f_i)$ is an *equidimensional variety of dimension* $n-i$). The *(geometric) affine degree* of $V(f_1, \ldots, f_i)$ is defined as usual as the cardinality of the finite set of points we obtain cutting $V(f_1, \ldots, f_i)$ by $n-i$ generic affine hyperplanes of $\mathbf{A}^n$ (more generally, we define the affine degree of a closed Zariski subset of $\mathbf{A}^n$ as the sum of the degrees of its irreducible components. See e.g. [25] or [18] for this notion of degree and its motivation). We denote the affine degree of $V(f_1, \ldots, f_i)$ by $degV(f_1, \ldots, f_i)$.

Let us now define $\delta := max\{degV(f_1, \ldots, f_i)\ ;\ 1 \le i \le n\}$ as the *(geometric) affine degree of the equation system* $f_1, \ldots, f_n$. We write $V := V(f_1, \ldots, f_n)$ and observe that V contains only finitely many, namely $degV \le \delta$ points.

Let us now consider the *toric case*. Fix $1 \le i \le n$. We define the *toric irreducible components* of the affine variety $V(f_1, \ldots, f_i)$ as those which are not contained in one of the hyperplanes $V(X_j)$, $1 \le j \le n$, of $\mathbf{A}^n$. We suppose that $f_1, \ldots, f_i$ form a *toric complete intersection*. This means that toric irreducible components of $V(f_1, \ldots, f_i)$ exist and have all dimension $n-i$. The *(geometric) toric degree* of $V(f_1, \ldots, f_i)$ is denoted by $deg^*V(f_1, \ldots, f_i)$ and is defined as the sum of the affine degrees of all toric components of $V(f_1, \ldots, f_i)$. We call $\delta^* := max\{deg^*V(f_1, \ldots, f_i)\ ;\ 1 \le i \le n\}$ the *(geometric) toric degree of the equation system* $f_1, \ldots, f_n$.

Let be given a non–zero linear form H of $k[X_1, \ldots, X_n]$ represented by its coefficients and let T be a new variable. We consider the straight–line program β and the coefficient n–tuple of H as the fixed input of all our principal algorithms. With these notations and assumptions, we may state our main results as follows :

Theorem 1 (the affine case). *Assume that the polynomials $f_1, \ldots, f_n$ form a regular sequence in $k[X_1, \ldots, X_n]$. Then there exists an arithmetic network with FOR gates which, from the input given by β and H, computes the coefficients of a non–zero polynomial $p \in k[T]$ such that $p(H)$ vanishes on $V = V(f_1, \ldots, f_n)$. The nonscalar size and depth of the network are $(nd\delta L)^{O(1)}$ and $O(n(log_2(\delta d) + \ell))$ respectively.*

Theorem 2 (the toric case). *Assume that for $1 \le i \le n$ the polynomials $f_1, \ldots, f_i$ form a toric complete intersection. Then there exists an arithmetic network with FOR gates which, from the input given by β and H, computes* *the coefficients of a non–zero polynomial $p^* \in k[T]$ such that $p^*(H)$ vanishes on the toric variety $V^* := V \setminus V(\prod_{i=1}^{n} X_i)$. The nonscalar size and depth of the network are $(nd\delta^* L)^{O(1)}$ and $O(n(log_2(\delta^* d) + \ell))$ respectively.*

Let us remark that the algorithms underlying Theorems 1 and 2 make substantial use of linear algebra subroutines dealing with matrices of size at most $2d\delta$ (affine case) and $2d\delta^*$ (toric case).

A natural question (raised by one of the referees) is the following : what "real life" complexities are hidden behind the notions of arithmetic network and straight–line program with FOR gates? What is their link to "ordinary" arithmetic networks and straight–line programs? For answering this question let us restrict ourselves to the affine case (the toric case can be discussed analogously).

A simple minded translation of the arithmetic network with FOR gates underlying Theorem 1 would produce an ordinary arithmetic network of nonscalar size and depth $(nd\delta^n L)^{O(1)}$ and $O(n(log_2(d\delta) + \ell))$ respectively. Whereas the nonscalar depth is fine, the nonscalar size of the network grows up to order of the sequential time complexity of usual Groebner basis computations for the elimination task we are considering, i.e. to $(nd^{n^2})^{O(1)}$ in worst case (here we assume that $\delta = d^n$ and $L = \binom{d+n}{n} = O(d^n)$ holds; see [7] and [14]).

The problem arising in this way can be localized in the iterative character of our algorithm (n iterations) which in the ordinary arithmetic network version produces stupidly growing straight–line programs for the representation of intermediate results. This is due to its recursive use of FOR calls. Applying "data compression" by means of the already mentioned homotopic deformation method one can lower the ordinary network size to $(nd^n)^{O(1)}$ which is an already known complexity bound for the elimination task we are considering ([9], [10], [21], [34], [36]. See also [2], [11], [27], [37]). This means the advantage of our method is reduced to exegetic "practical complexity" as far as running time is concerned (nevertheless we may have saved something with respect to storage space).

However there are good news : a more radical compression of expanding straight–line programs during the execution of the algorithms is possible thanks to a symbolic adaptation of Newton's method (or Hensel iteration as someones like to call it in case of positive characteristic) in combination with "Vermeidung von Divisionen" ([48]). The final outcome is the following result :

Theorem 3. *Assume that $f_1, \dots, f_n$ form a regular sequence in $k[X_1, \dots, X_n]$ and that for each $1 \leq i \leq n$ the ideal $(f_1, \dots, f_i)$ is reduced. Then there exists an (ordinary) arithmetic network which, from the input given by β and H, computes the coefficients of a non–zero polynomial $p \in k[T]$ such that $p(H)$ vanishes on $V = V(f_1, \dots, f_n)$. The network uses parameters from the algebraic closure of the prime field of k and has nonscalar size and depth $(nd\delta L)^{O(1)}$ and $O(n(log_2(\delta.d) + \ell))$ respectively.*

Of course the toric counterpart of Theorem 3 is valid too. In this case (and also in case of Theorem 2) the parameters L and ℓ measuring nonscalar size and depth of the input circuit β can be replaced by the sparsity N of the polynomials $f_1, \ldots, f_n$ and by $log_2 d$ respectively (the sparsity is the number of monomials with non–zero coefficients appearing in $f_1, \ldots, f_n$). The nonscalar size and depth of the resulting network are then $(nd\delta^* N)^{O(1)}$ and $O(nlog_2(d\delta^*))$ respectively. For lack of space in these proceedings we do not include the complete proof of Theorem 3 in the paper. However the proof of Theorem 1 we give here represents an essential part of the proof in question of Theorem 3 which will be given in the full paper version of this contribution.

The key point of our method consists in a consequent use of problem adapted data structures : arithmetic networks and straight–line programs with (and without) FOR gates. A similar ground idea may be found in the impressive theoretic and practic work of D. Duval and her school on dynamic evaluation ([16]). Closest to our complexity results comes the numeric method of M. Shub and S. Smale for finding "approximate" solutions of zero–dimensional homogeneous "average" equations systems ([45]). Their method too is independent on the Bézout number and the regularity of the input ideal. The sequential time complexity of their algorithm (measured as ours by counting just arithmetic operations) is polynomial in $n\binom{d+n}{n}$ (the size of the densely given input system) and the number of approximate solutions wanted. However caution is necessary in the interpretation of their result as "equation solving" in the sense of computer algebra. First they need sufficient genericity of the system (just the contrary of the point of view of computer algebra which focusses on "special systems"). Secondly Newton iteration (for which they seek approximate solutions) is inefficient in terms of bit–complexity if the rational numbers appearing in it are given in binary coding. This fact is due to Liouville's estimate (see [41]). However this second drawback can be avoided coding rational numbers by straight–line programs. Identity of rational numbers given by such codings can be detected in random polynomial time by a BPP–test ([30]) or a fixed sample test (as in [28] or [12]) which is due to [24].

Our paper is organized as follows. In Section 2 we introduce our computational model of arithmetic networks and straight–line programs with (and without) FOR gates. In Section 3 we review some known technical lemmas which are necessary for the proofs in Section 4. The essential steps of the proof of Theorem 1 and 2 are contained in Section 4. This proof is based on a recursive construction of a Noether normalization and a primitive element for the varieties $V(f_1, \ldots, f_i)$, $1 \leq i \leq n$, followed by a procedure of cleaning extraneous irreducible components. These recursive constructions are explained in Section 4 and a recursion law of the complexity of the underlying algorithm is given.

2 The Computational Model

The idea of using straight–line programs as succinct encodings of special polynomials (as they appear in elimination problems) goes back to the late seventies and was discovered by different people independently (one of them being J. Morgenstern who preferred oral "hadise" to hardcover publications for the dissemination of his thought). This idea appears implicitly or explicitly in the following representative (although not exhaustive) list of papers, mostly dedicated to probabilistic testing of polynomial identities : [28], [30], [44], [13], [49]. First applications of this point of view to computer algebra dealt only with the elimination of just one variable (absolute primality testing, greatest common divisor computation and factorisation of multivariate polynomials; see [29], [31], [32]) and were later extended to multivariate elimination problems by means of "duality techniques" of different type and "Vermeidung von Divisionen" ([17], [21], [22], [34], [35]; see also [27]).

Straight–line programs and arithmetic circuits have their origins in the design of semi–numerical algorithms and build a link between numerical analysis and computer science (see [20], [26], [47] and the references given there). In this section, we slightly modify the notion of "ordinary" straight–line programs and arithmetic networks adding a special type of computation nodes : FOR gates. These gates modelize something which can be paraphrased as "parallel FORTRAN programs".

A piecewise rational function is a mapping $\varphi : \mathbf{A}^n \longrightarrow \mathbf{A}^1$ such that there exists a partition of $\mathbf{A}^n$ in k–definable constructible subsets $\{C_i : 1 \leq i \leq M\}$, such that for every $1 \leq i \leq M$, there is a rational function $\phi_i \in k(X_1, \ldots, X_n)$ defined everywhere on C_i, which verifies :

$$\varphi \mid_{C_i} = \phi_i \mid_{C_i}$$

An ordinary arithmetic network is a device that evaluates piecewise rational functions. However, in our applications, most of the functions are piecewise polynomial, i.e. the ϕ_i are polynomials belonging to $k[X_1, \ldots, X_n]$.

An arithmetic network Γ over k is a pair $\Gamma = (\mathcal{G}, Q)$, where $\mathcal{G}$ is directed acyclic graph, with vertices (called nodes or gates of Γ) of indegree 0, 1, 2 or 3 and where Q is an assignment (labeling) of instructions and (piecewise) rational functions ("intermediate results") to nodes which will be specified later. In $\mathcal{G}$ there are n nodes of indegree 0, labeled by the variables $X_1, \ldots, X_n$ and called the input nodes of Γ.

We define the depth of a node ν of the graph $\mathcal{G}$ as the length of the longest path joining ν with some input gate. Let us denote any node of $\mathcal{G}$ by a pair of integers (i, j), where i represents the depth of the node and j is its ordinal number given by some ordering of the set of nodes of depth i (see [34], [40] for this type of encoding of arithmetic networks).

Let (i, j) be a node of $\mathcal{G}$. If (i, j) has indegree 0 and does not represent an input, the labeling Q assigns to this node a constant from k (which we call subsequently a *parameter* of Γ). If (i, j) has indegree 2 the labeling Q

assigns to it an arithmetic operation of k. If (i,j) has indegree 1 the labeling Q assigns to it a *sign* (i.e. equal to zero) *test* and a *boolean variable.* If (i,j) has indegree 3 the labeling Q assigns to it a *selector* which makes a choice between two prefixed nodes of $\mathcal{G}$ according to the value of a boolean variable associated with a third node. A node of $\mathcal{G}$ labeled in this manner is called a *gate* of Γ. We have already introduced the input gates. A gate which is labeled by a constant or an arithmetic operation of k is called a *computation gate.* The other gates are called *sign* or *selector gates*, depending on their labeling. We suppose that the labeling Q is "meaningful", i.e. computation gates have ingoing edges (if any) coming from computation or input gates, sign gates have an ingoing edge coming from a computation gate , and selector gates have two ingoing edges coming from computation or input gates and another ingoing edge coming from a sign gate. Under this hypothesis the labeling Q associates in a obvious way to each computation gate (i,j) a rational function which we denote by $Q_{i,j}$. Let (i,j) be a sign gate with ingoing edge coming from a node (r,s) and let x be a point of $\mathbf{A}^n$ where $Q_{r,s}$ is defined. Then the boolean variable $B_{i,j}$ associated to (i,j) values the value 1 or 0 according to the truth or falseness of the statement "$Q_{i,j}(x) = 0$". If (i,j) is a selector gate with two ingoing edges coming from computation gates (k,l) and (k',l') and if x is a point of $\mathbf{A}^n$ where $Q_{k,l}$ and $Q_{k',l'}$ are defined and if the third ingoing edge of (i,j) comes from the sign gate (r,s) then we associate to (i,j) the field element $Q_{k,l}(x)$ or $Q_{k',l'}(x)$ according to the value which takes in x the boolean variable $B_{r,s}$ associated to the sign gate. We consider the gates of Γ with outdegree 0 as *outputs* and we suppose that for each $x \in \mathbf{A}^n$ there exists a "stream" of "consistent" paths (with all rational functions defined in x) to a prefixed number of output gates.

A computation gate of Γ is called *nonscalar* if the instruction associated to it corresponds to a multiplication of non–constant rational functions or to a division by a non–constant rational function. We assign to nonscalar computation gates unit cost whereas all other gates are taken cost–free (in particular, computation gates of Γ corresponding to k–linear operations are free). In this way, we associate two complexity measures to the arithmetic network Γ :

- *nonscalar sequential time* or the *nonscalar size* of Γ, defined as the total number of non–scalar computation gates of Γ,
- *nonscalar parallel time* or the *nonscalar depth* of Γ defined as the longest oriented path of Γ joining an input gate with an output gate when only nonscalar computation gates count for the "length".

These two nonscalar complexity measures are fairly realistic with respect to operation counting because Γ can always be rearranged such that the *total* number of gates and the *total* depth of Γ are bounded roughly by the square of its nonscalar size and depth. The intuitive meaning of the nonscalar size of Γ is (sequential) running time whereas the nonscalar depth is linked to rather mathematical quantities as degree and height (in case $k := \mathbb{Q}$) of the rational functions appearing as intermediate results of Γ. The total depth of Γ has a natural interpretation as minimal storing space (see [34] for more details on this subject).

We extend now this model of "ordinary" arithmetic gates by adding a new type of computation nodes, called *FOR gates*. A FOR gate (i,j) has indegree 1 and outdegree N, where N is an arbitrary natural number. Associated to it there exist fixed values $\eta_1,\ldots,\eta_N \in k$ (e.g $\eta_1 = 1,\ldots,\eta_N = N$ in case $k := \mathbb{Q}$) and a fixed index $1 \leq k \leq n$ pointing to the variable X_k. The unique ingoing edge of (i,j) comes from a computation or a selector gate (r,s). Thus the FOR gate receives a rational function $Q_{r,s}$ of $k(X_1,\ldots,X_n)$ which in all our applications will be a polynomial, hence everywhere defined in $\mathbf{A}^n$. The gate (i,j) specializes the variable X_k in $Q_{r,s}$ to the values $\eta_1,\ldots,\eta_N$ and puts the results (if defined) on its outgoing edges. In our nonscalar model we count FOR gates by the cost "number of outgoing edges". (This sounds temerarious from the point of view of "real computing". Nevertheless the defect is remedied in our applications by the algorithm underlying Theorem 3). We shall refer to networks which include FOR instructions as "arithmetic networks with FOR gates".

A special case of arithmetic networks with (and without FOR gates) are those which have as single output a boolean combination of sign gates. We call them *decision networks.* Another important case of special networks is represented by those which contain only computation gates (no sign or selector gates). They are called *straight–line programs* or *arithmetic circuits* and they compute (or represent) rational functions of $k(X_1,\ldots,X_n)$. Often we call the size of a straight–line program its *length.*

An arithmetic network without any nonscalar division (only divisions by non–zero elements of k are allowed in this case) is called *division free.* Division free straight–line programs compute polynomials of $k[X_1,\ldots,X_n]$. Finally we say that a family of arithmetic networks with nonscalar sequential time cost function L is *well parallelizable* if its nonscalar parallel time cost function ℓ behaves as $\ell = O(log_2 L)$.

Since we are not considering other complexity measures than the nonscalar ones, we simplify our terminology from mow on saying just *size/sequential* time or *depth/parallel* time with reference to the nonscalar cost measure.

3 Technical Lemmas

3.1 Squarefree Representation and Greatest Common Divisor Computation of Univariate Polynomials

Let R be a factorial domain with fraction field K, T a new variable, D a fixed natural number and $p,q \in R[T]$ two polynomials of formal degree D. We think that p and q are given by their coefficient vectors each of length D. In the statements which follow now we refer to (ordinary) arithmetic networks and straight–line programs over R which receive the coefficient vectors of p and q as inputs. Furthermore greatest common divisors (gcd's) will always be taken with respect to the principal ideal domain $K[T]$ although they are represented in $R[T]$ (thus they are not unique up to units in R). The proof of the following well-known lemmas can be found in [34].

Lemma 4. *There exists a division free well parallelizable arithmetic network Γ of size $O(D^6)$ which from the coefficients of p and q computes the coefficients of a greatest common divisor (in $R[X]$) of these polynomials.*

The greatest common divisor computed by the algorithm underlying this Lemma is a (not necessarily primitive) polynomial of $R[T]$ which we denote by $gcd(p,q)$. A similar well parallelizable ($O(s^5D^6)$ sequential time) complexity bound holds for the computation of the simultaneous greatest common divisor of more than two (say s) polynomials of degree at most D belonging to $R[T]$.

Sometimes the second polynomial q is given by a division free straight–line program in $R[T]$ of length L and depth ℓ instead by its coefficients. In this case we have

Lemma 5. *There exists a division free arithmetic network of size $LD^{O(1)}$ and depth $O(log_2(D)+\ell)$ which computes the coefficients of the greatest common divisor of p and q .*

The algorithm underlying Lemma 5 uses linear algebra subroutines which deal with square matrices over R of size at most $2D-1$.

As before this lemma can be generalized to the case of the computation of the simultaneous greatest common divisor of more than two (say s) polynomials, one of them given by its coefficient vector and having degree at most D and the others given by a division free straight–line program in $R[T]$ of length L and depth ℓ . The outcome is then a division free arithmetic network of size $(L+s)D^{O(1)}$ and depth $O(log_2(D)+\ell)$ which computes the coefficients of the greatest common divisor in question. Again the linear algebra subroutines deal only with square matrices of size at most $2D-1$. Lemma 4 can be used to compute the squarefree representation of the polynomial p if its derivative p' is non-zero (in our applications we may always suppose that, even in case of positive characteristic; see [21] and [22] for details). *So let us assume* $p' \neq 0$. Observe that the coefficients of p' are immediately obtained from the (given) coefficients of p . Therefore we have

Lemma 6. *There exists a division free well parallelizable arithmetic network of size $O(D^6)$ which from the coefficients of p computes the coefficients of a univariate polynomial $p^* \in R[T]$ and a non–zero element of the ring $\theta \in R$, such that :*

$$p^* = \theta \frac{p}{gcd(p,p')}$$

3.2 Correct Test Sequences and "Vermeidung von Divisionen"

The algorithms in [17], [21], [22], [34], [35] rely heavily on the use of "correct test sequences" ([12], [28], [34]) as a tool for deciding identity of polynomials given by straight–line programs. Unlike the probabilistic identity tests [13], [30], [44], [49] the choice of a suitable correct test sequence does not depend on the specific polynomials whose identity has to be checked but only on the number of variables and the size of the input circuit. The outcome are non–uniform deterministic or random algorithms of a stronger type as those proposed in [44] and [49].

Definition 7. Let $\mathcal{F}$ be a set of polynomials of $k[X_1, \ldots, X_n]$ such that 0 belongs to $\mathcal{F}$. Let $\mathcal{Q}$ be a subset of k^n. $\mathcal{Q}$ is called a correct test sequence (or questor set) for $\mathcal{F}$ if for any $P \in \mathcal{F}$ the following implication holds :

$$P(x) = 0 \; for \; all \; x \in \mathcal{Q} \Longrightarrow P = 0 \; .$$

The cardinality $\#\mathcal{Q}$ is called the *length* of the test sequence $\mathcal{Q}$.

The existence of short correct test sequences is warranted by the following fact :

Lemma 8. *Let $\mathcal{F}$ be the class of all polynomials of $k[X_1, \ldots, X_n]$ which can be evaluated by a straight–line program in $k(X_1, \ldots, X_n)$ of (nonscalar) size L and depth ℓ. Let $\omega := (2^{\ell+1} - 2)(2^{\ell} + 1)^2$* and *$\sigma := 6(\ell L)^2$. Then for any collection Ω of ω elements of k the set $\Omega^n \subset k^n$ contains at least $\omega^{n\sigma}(1 - \omega^{-\frac{\sigma}{6}})$ correct test sequences of length σ for the class $\mathcal{F}$.*

If the characteristic of k is zero we shall always make the standard choice $\Omega := \{1, \ldots, \omega\} \subset \mathbb{Z}$.

From the proof ([28], [34]) of Lemma 8 we deduce that its conclusion remains true if we take as $\mathcal{F}$ the class of all polynomials of $k[X_1, \ldots, X_n]$ which can be evaluated by straight–line programs *with FOR gates* of size L and depth ℓ. Lemma 8 implies that for $L > 0$ and $\ell > 0$ correct test sequences for $\mathcal{F}$ of length σ polynomial in L always exist, and that any random choice of σ elements in Ω^n leads to a correct test sequence for $\mathcal{F}$ with an error probability of $\omega^{-\frac{\sigma}{6}} << \frac{1}{2}$.

We shall make frequent use of a method called "Vermeidung von Divisionen" which is due to [48] and which allows to transform any straight–line program Γ (with or without FOR gates) which computes a polynomial and contains essential divisions into an equivalent division free one, say Γ'. If the size and the depth of Γ are L and ℓ respectively the the size and depth of Γ' are D^2L and $O(\ell)$ (see [34] for the proof of this precise statement).

3.3 Construction of a Primitive Element

Primitive element constructions (or "shape lemmas") are crucial for solving zero–dimensional polynomial systems symbolically (see [1], [9], [10], [11], [21], [23], [27], [34], [36], [39]). They furnish a handy description of the multiplication tensor of the finite dimensional $k - algebras$ which arise as (reduced or non-reduced) coordinate rings of such equation systems.

Let R be an integral domain with fraction field K. Without loss of generality (see [21]) we assume K to be of characteristic zero. Let I be a zero-dimensional ideal of $K[X_1, \ldots, X_n]$. We denote the radical of I by $\sqrt{I}$.

For any maximal ideal $\mathcal{Q}$ associated to I, let $K(\mathcal{Q})$ be the field $K(\mathcal{Q}) = K[X_1, \ldots, X_n]/\mathcal{Q}$. A linear form $u = t_1X_1 + \cdots + t_nX_n \in R[X_1, \ldots, X_n]$ is said to be a *primitive element* for I (or the finite dimensional $K-algebra$ $K[X_1, \ldots, X_n]/I$) if it satisfies the following two conditions :

- for any maximal ideal $\mathcal{Q}$ associated to I, the minimal equation (minimal polynomial) of u modulo $\mathcal{Q}$ has degree $\delta(\mathcal{Q}) := [K(\mathcal{Q}) : K]$.
- For any two different maximal ideals $\mathcal{Q}$ and $\mathcal{Q}'$ associated to I, the minimal polynomials of u modulo $\mathcal{Q}$ and $\mathcal{Q}'$ are distinct.

We observe that a linear form $u \in R[X_1, \ldots, X_n]$ represents a primitive element of I if and only if the set $\{1, u, u^2, u^3, \ldots\}$ generates modulo I the $K-$vector space $K[X_1, \ldots, X_n]/I$.

For the construction of a primitive element from a given set of generators of I we follow the lines of [21] or [34].In order to explain our method we introduce new variables $T_1, \ldots, T_n$ and for $1 \leq j \leq n$ we consider the following domains :

$$R_j := R[T_1, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n]$$

$$K_j := K(T_1, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n),$$

and the following linear form in $X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_n$:

$$Z_j := T_1X_1 + \cdots + T_{j-1}X_{j-1} + T_{j+1}X_{j+1} + \cdots + T_nX_n.$$

Furthermore we consider the following linear form U in $X_1, \ldots, X_n$:

$$U := T_1X_1 + \cdots + T_nX_n.$$

We observe that for any $1 \leq j \leq n$ the identity $U = Z_j + T_jX_j$ holds. Let T be an additional variable and D a natural number. The way how we shall generate a primitive element is contained in the following statement :

Lemma 9. *Let be given the following inputs :*

- *monic polynomials $f_1, \ldots, f_n \in R[T]$ of degree at most D such that for each $1 \leq j \leq n$ the specialization $f_j(X_j)$ belongs to the ideal I.*
- *for each $1 \leq j \leq n$ a polynomial $g_j \in R_j[T]$ which is monic in T and which has total degree at most D such that the specialization $g_j(Z_j)$ belongs to the ideal $K_j \bigotimes_K I$.*

We suppose further that $f_1, \ldots, f_n$ and $g_1, \ldots, g_n$ are represented by their coefficients with respect to the variable T. Moreover we assume that the coefficients of $g_1, \ldots, g_n$ themselves are given by a division free straight–line program in $R[T_1, \ldots, T_n]$ of length L and depth ℓ.

Then there exists a division free arithmetic network in R of size $(nDL)^{O(1)}$ and depth $O(log_2(nD) + \ell)$ which computes the coefficient representation of the following items :

1. *a primitive element* $u = \lambda_1 X_1 + \cdots + \lambda_n X_n \in \mathbb{Z}[X_1, \ldots, X_n]$ *for the ideal* I,
2. *a non–zero element* $\rho \in R$ *and for each* $1 \leq j \leq n$ *a polynomial* $v_j \in R[T]$ *such that* $\rho X_j - v_j(u) \in \sqrt{I}$ *holds,*
3. *a monic squarefree polynomial* $q \in R[T]$ *such that* $q(u) \in \sqrt{I}$ *holds.*

For $1 \leq i \leq n$ *we have* $deg v_i < deg q$ *and* $deg q$ *equals the degree of the affine variety defined by* I *over the algebraic closure of* K.

Lemma 9 is contained in [34], Proposition 27 and its proof (see also [21], Sections 3.4.6 and 3.4.7).

4 The Algorithm

In this Section we describe the algorithmic procedure underlying Theorems 1 and 2.

Let us fix the following notions and notations : let X_0 be a new variable. For any non–zero polynomial $p \in k[X_1, \ldots, X_n]$ of degree D we define its homogenization ${}^h p$ (with respect to the variable X_0) as

$${}^h p := X_0^D p(X_1/X_0, \ldots, X_n/X_0).$$

For $p = 0$ we put ${}^h p := 0$. Of course ${}^h p$ is a homogeneous polynomial in $k[X_0, \ldots, X_n]$ of degree D. For an ideal I of $k[X_1, \ldots, X_n]$, we define its homogenization ${}^h I$ as the (homogeneous) ideal in $k[X_0, \ldots, X_n]$ generated by the set of polynomials $\{{}^h f \ : f \in I\}$.

We maintain assumptions and notations of Section 1. In particular we suppose to be given input polynomials $f_1, \ldots, f_n \in k[X_1, \ldots, X_n]$ of degree at most d which are encoded by a division free straight–line program of size L and depth ℓ. We suppose that $f_1, \ldots, f_n$ form an affine or toric complete intersection, according to the problem we are considering. Let H be a non–zero linear form of $k[X_1, \ldots, X_n]$ given by its coefficients. The problem we want to solve is the following : find a non–zero polynomial p (in the affine case) or a non–zero polynomial p^* (in the toric case) belonging to $k[T]$ such that this polynomial vanishes on $V = V(f_1, \ldots, f_n)$ (in the affine case) or on $V^* = V(f_1, \ldots, f_n) \setminus V(\prod_{i=1}^{n} X_i)$ (in the toric case).

From [34] Lemma 13 we deduce that the homogeneous polynomials ${}^h f_1, \ldots, {}^h f_n$ can be evaluated by a division free straight–line program in $k[X_0, \ldots, X_n]$ of (nonscalar) size and depth $d(d+1)^2 L$ and $log_2 d + 2\ell$ respectively.

For $1 \leq i \leq n$ we denote by $I_i := (f_1, \ldots, f_i)$ the ideal generated by $f_1, \ldots, f_i$ in $k[X_1, \ldots, X_n]$. Let us now introduce the specific notions and notations which we need in the *affine case*. Fix $1 \leq i \leq n$ and let $J_i := \sqrt{{}^h I_i}$ be the radical of the homogenization of $I_i = (f_1, \ldots, f_i)$. Thus J_i is a homogeneous and unmixed radical ideal of codimension i and the corresponding projective variety does not contain any irreducible component at infinity. Furthermore the homogeneous polynomial ${}^h f_{i+1}$ is a non–zero divisor modulo J_i for $i < n$. Let $\mathcal{I}_i := ({}^h f_i, J_{i-1})$ be the homogeneous ideal generated by ${}^h f_i$ and J_{i-1}. We observe that J_i is the intersection of all codimension i prime homogeneous ideals which contain $\mathcal{I}_i$ but not the form X_0 (of course, these are associated primes of $\mathcal{I}_i$).

Let us finally consider the specific notions and notations of the *toric case*. Let $1 \leq i \leq n$. We denote by J_i^* the intersection of all homogeneous prime ideals of codimension i which contain ${}^h I_i$ but not the form $\prod_{i=0}^{n} X_i$ (these prime ideals are again associated to ${}^h I_i$). By hypothesis (the family $f_1, \ldots, f_n$ forms a toric complete intersection) the variety $V(I_i)$ is not contained in $V(\prod_{i=1}^{n} X_i)$ and therefore $J_i^* \neq (X_0, \ldots, X_n)$. Furthermore the homogeneous polynomial ${}^h f_{i+1}$ is not a zero divisor modulo J_i^* for $i < n$. Finally, as in the affine case, J_i^* is the intersection of all codimension i homogeneous (associated) prime ideals which contain $({}^h f_i, J_{i-1}^*)$ but not the form $\prod_{i=0}^{n} X_i$.

With these notions and notations fixed we are now able to describe the principal items which are produced by the algorithm underlying Theorems 1 and 2 as intermediate results and to indicate their main properties.

For the affine as well as for the toric case the algorithm proceeds in an analogous manner by n recursive steps. We give now a simultaneous account of the i–step of the algorithm both for the affine and the toric case (here $1 \leq i \leq n$). The algorithm produces first a linear change of the variables $X_0, \ldots, X_n$ into new variables $Y_0, \ldots, Y_n$ such that the following canonical ring homomorphism becomes an integral extension :

– *In the affine case :*

$$k[Y_0, \ldots, Y_{n-i}] \longrightarrow k[Y_0, \ldots, Y_n]/J_i$$

– *In the toric case :*

$$k[Y_0, \ldots, Y_{n-i}] \longrightarrow k[Y_0, \ldots, Y_n]/J_i^*$$

Next the algorithm generates a non–zero linear form $u_i \in k[Y_{n-i+1}, \ldots, Y_n]$ and a homogeneous polynomial of $k[Y_0, \ldots, Y_{n-i}, T]$ which is monic with respect to the variable T and which we denote by q_i in the affine and by q_i^* in the toric case. The linear form u_i is a primitive element for the ideal J_i in the affine and for J_i^* in the toric case having minimal polynomial q_i or q_i^* respectively. Finally the algorithm produces non–zero homogeneous polynomials $\rho_i \in k[Y_0, \ldots, Y_{n-i}]$ and $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)} \in k[Y_0, \ldots, Y_{n-i}, T]$ such that in the localized ring $k[Y_0, \ldots, Y_{n-i}]_{\rho_i}[Y_{n-i+1} \ldots, Y_n]$ the ideal generated by

$$q_i(u_i), \rho_i Y_{n-i+1} - v_{n-i+1}^{(i)}(u_i), \ldots, \rho_i Y_n - v_n^{(i)}(u_i)$$

is identical with the ideal (J_i) generated by J_i (affine case). A similar statement is valid for the toric case if we replace q_i by q_i^* and J_i by J_i^*.

Moreover the degree of $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)}$ in T is strictly less than the degree of q_i or q_i^* respectively (this latter condition makes $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)}$ unique with respect to $k(Y_0, \ldots, Y_{n-i})[T]$).

Thus the $k(Y_0, \ldots, Y_{n-i})-$ algebra $k(Y_0, \ldots, Y_{n-i})[T]/(q_i)$ (or equivalently $k(Y_0, \ldots, Y_{n-i})[u_i]/(q_i(u_i))$) is isomorphic to $k(Y_0, \ldots, Y_{n-i})[Y_{n-i+1}, \ldots, Y_n]/J_i$ in the affine case (and the isomorphism can be effectively computed). An analogous statement is true in the toric case.

All items (namely u_i , q_i , q_i^* , ρ_i , $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)}$) are given by its coefficients with respect to the variables $Y_{n-i+1}, \ldots, Y_n, T$ and these coefficients (which belong to $k[Y_0, \ldots, Y_{n-i}]$) are represented by a (generally division free) straight–line programs with FOR gates in $k[Y_0, \ldots, Y_{n-i}]$. Furthermore, our procedure produces these straight–line programs.

Finally let us observe that these items which appear as intermediate results of our procedure are canonical and intrinsic objects with a precise geometric meaning (on just this meaning is based the data compression which leads to Theorem 3). Let us consider only the affine case : to the linear change of the variables $X_0, \ldots, X_n$ into $Y_0, \ldots, Y_n$ corresponds a finite surjective morphism of affine varieties $\pi_i : V(f_1, \ldots, f_i) \longrightarrow \mathbf{A}^{n-i}$. The linear form u_i complements π_i to a finite morphism (π_i, u_i) which maps $V(f_1, \ldots, f_i)$ onto a hypersurface of $\mathbf{A}^{n-i+1}$. The minimal equation of this hypersurface is just the polynomial we obtain specializing in $q_i \in k[X_0, \ldots, X_n]$ the variable X_0 to 1 . With the same type of specialization we see that ρ_i describes the discriminant variety of the generic fiber of π_i and that the polynomials $v_{n-i}^{(i)}, \ldots, v_n^{(i)}$ represent a "univariate" rational parametrization of the π_i–fibers of the points of $\mathbf{A}^{n-i}$ which lie outside of this discriminant variety.

4.1 The Recursion

Keep $1 \leq i < n$ and notions and notations of the last subsection fixed. We are going to describe the $i+1$–th step of our recursive main algorithm. Let $\delta_i := degV(f_1, \ldots, f_i)$ and $\delta_i^* := deg^*V(f_1, \ldots, f_i)$ and recall that $degf_j \leq d$ holds for $1 \leq j \leq n$. With these notations we have $\delta_i = deg_T q_i = degq_i$, $\delta_i^* = deg_T q_i^* = degq_i^*$. Let Γ_i be a division free straight–line program with FOR gates in $k[Y_0, \ldots, Y_{n-i}]$ which evaluates the coefficients with respect to $Y_{n-i+1}, \ldots, Y_n$ and T of the polynomials u_i , ρ_i , $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)}$ and of q_i in the affine, and of q_i^* in the toric case. Denote by Λ_i and λ_i the (nonscalar) size and depth of Γ_i respectively.

The core of our procedure is the following technical result :

Proposition 10. *There exists an arithmetic network with FOR gates in* $k[Y_0, \ldots, Y_{n-i}]$ *of size* $(nd\delta L)^{O(1)}$ *and depth* $O(log_2(nd\delta) + \ell)$ *which from* Γ_i *as input produces a linear change of variables* $Y_0, \ldots, Y_n$ *into new variables, say* $Y_0', \ldots, Y_n'$*, such that* $Y_0', \ldots, Y_n'$ *are in Noether position with respect to the ideal* J_{i+1} *in the affine case and to* J_{i+1}^* *in the toric case, the variables* $Y_0', \ldots, Y_{n-1-1}'$ *being free.*

Furthermore the network produces the coefficients of a linear form $u_{i+1} \in k[Y'_{n-i}, \ldots, Y'_n]$ *and a division free straight–line program with FOR gates in* $k[Y'_0, \ldots, Y'_{n-i-1}]$ *which represents the polynomial* ρ_{i+1} *and the coefficients with respect to* T *of* $v^{(i+1)}_{n-i}, \ldots, v^{(i+1)}_n \in k[Y'_0, \ldots, Y'_{n-i-1}, T]$ *and of* q_{i+1} *or* $q^*_{i+1} \in k[Y'_0, \ldots, Y'_{n-i-1}, T]$ *in the affine and toric case respectively.*

The nonscalar size and depth of this straight–line program are $(id\delta_i L)^{O(1)} + \Lambda_i$ *and* $O(log_2(id\delta_i) + \ell) + \lambda_i$ *respectively.*

We observe without proof that the algorithm underlying this procedure can be organized in such a way that it uses only linear algebra subroutines dealing with square matrices of size at most $2d\delta_i$ (or $2d\delta^*_i$).

We divide the $i+1$–th recursive step of our algorithm in three parts :

– *recursive Noether normalization.*
– *recursive generation of a primitive element.*
– *cleaning extraneous irreducible components.*

The first two parts are common for both the affine and the toric case. Only the third part distinguishes in some few technical points between the two cases.

Before giving the details of these three parts, let us recall the following procedure to compute resultants which is implicitly contained in [7] and [43] :

Lemma 11. *Let* J *be an unmixed homogeneous radical ideal of* $k[X_0, \ldots, X_n]$ *and let us suppose that the canonical homomorphism*

$$k[X_0, \ldots, X_{n-i}] \longrightarrow k[X_0, \ldots, X_n]/J.$$

is an integral graded ring extension. Then, for an homogeneous polynomial $g \in k[X_0, \ldots, X_n]$ *its minimal polynomial* $m_g \in k[X_0, \ldots, X_{n-i}, T]$ *has the form*

$$T^D + a_{D-1}T^{D-1} + \cdots + a_0,$$

where $D \leq degV(J)$ *and* $a_j \in k[X_0, \ldots, X_{n-i}]$ *is zero or a homogeneous polynomial of degree* $(D-j) \cdot deg(g)$ *.*

The same conclusion remains true if we replace the minimal polynomial m_g *by the characteristic polynomial* χ_g *of* g *. Moreover the minimal polynomial* m_g *is squarefree.*

Proof. Observe first that the assumption J unmixed implies that the quotient ring $k[X_0, \ldots, X_n]/J$ is a free $k[X_0, \ldots, X_{n-i}]$–module of finite rank (this follows easily from an ad hoc argument based on the graded structure of $k[X_0, \ldots, X_n]/J$ or more generally from the Quillen–Suslin theorem). Thus m_g and χ_g belong to $k[X_0, \ldots, X_{n-i}, T]$.

Let $J = \bigcap_{j=1}^{N} \mathcal{Q}_j$, where the $\mathcal{Q}_j$'s are the homogeneous prime ideals of codimension i associated to J . From our assumptions we obtain for $1 \le j \le N$ an integral ring extension :

$$k[X_0, \ldots, X_{n-i}] \hookrightarrow k[X_0, \ldots, X_n]/\mathcal{Q}_j .$$

The minimal polynomial $m_g^{(j)}$ of the homothety defined by the multiplication by g modulo $\mathcal{Q}_j$ satisfies the conclusions of the lemma (see [7], Remark 9). Since the minimal polynomial of the homothety defined by the multiplication by g modulo J is a product of some of the given $m_g^{(1)}, \ldots, m_g^{(N)}$, the first assertion of the lemma follows. The third one is a consequence of the fact that the polynomials $m_g^{(1)}, \ldots, m_g^{(N)}$ are irreducible and that J is radical. The second assertion follows from the observation that the irreducible factors of χ_g are the polynomials $m_g^{(1)}, \ldots, m_g^{(N)}$. □

Observe that if g is not a zero divisor modulo J , the term a_0 is a non–zero homogeneous polynomial of $k[X_0, \ldots, X_{n-i}]$ of degree $D \cdot deg(g)$.

Recursive Noether Normalization. Without loss of generality we describe just the affine case (the toric case follows simply by replacing J_i by J_i^* and so on).

First we observe that ${}^h f_{i+1}$ is not a zero divisor modulo J_i . According to the notations of Lemma 11 above let $m_{{}^h f_{i+1}}$ be the minimal polynomial of the homothety given modulo J_i by ${}^h f_{i+1}$:

$$m_{{}^h f_{i+1}} = T^D + a_{D-1} T^{D-1} + \cdots + a_0 .$$

The coefficient $a_0 \in k[Y_0, \ldots, Y_{n-i}]$ is a non-zero homogeneous polynomial of degree at most $d\delta_i$ and $m_{{}^h f_{i+1}}$ is squarefree by Lemma 11.

In order to obtain a Noether normalization with respect to $\mathcal{I}_{i+1} = ({}^h f_{i+1}, J_i)$ (and hence with respect to J_{i+1}) we have just to find a linear change of the variables $Y_0, \ldots, Y_{n-i}$ into new ones, say $Y_0', \ldots, Y_{n-i}'$, such that the (uniquely defined) polynomial $A_0(Y_0', \ldots, Y_{n-i}') \in k[Y_0', \ldots, Y_{n-i}'] = k[Y_0, \ldots, Y_{n-i}]$ which satisfies the equality

$$A_0(Y_0', \ldots, Y_{n-i}') = a_0(Y_0, \ldots, Y_{n-i})$$

is monic in the variable Y_{n-i}' . We obtain this linear change of variables as follows :

First observe that modulo J_i the forms $1, u_i, \ldots, u_i^{\delta_i - 1}$ represent a basis of the $k(Y_0, \ldots, Y_{n-i})$–vector space $k(Y_0, \ldots, Y_{n-i})[Y_{n-i+1}, \ldots, Y_n]/(J_i)$. By assumption the coefficients with respect to the variable T of the polynomials q_i , $v_{n-i+1}^{(i)}, \ldots, v_n^{(i)} \in k[Y_0, \ldots, Y_{n-i}, T]$ as well as the polynomial $\rho_i \in k[Y_0, \ldots, Y_n]$ are given by the circuit Γ_i . This implies that we can write down directly the (homogeneous) entries of the companion matrix $M_i \in k[Y_0, \ldots, Y_{n-i}]^{\delta_i \times \delta_i}$ of

q_i. The matrix M_i describes the homothety defined by the multiplication by u_i in $k(Y_0,\ldots,Y_{n-i})[Y_{n-i+1},\ldots,Y_n]/(J_i)$ with respect to the basis given by the forms $1,u_i,\ldots,u_i^{\delta_i-1}$. With respect to the same basis the homothety defined by the multiplication by ${}^hf_{i+1}$ has a matrix representation $M_{{}^hf_{i+1}}$ which satisfies the equation :

$$\rho_i^k M_{{}^hf_{i+1}} = {}^hf_{i+1}(Y_0,\ldots,Y_{n-i},v_{n-i+1}^{(i)}(M_i),\ldots,v_n^{(i)}(M_i))$$

for some $k\leq d$ (in a slight abuse of notation we write ${}^hf_{i+1}(Y_0,\ldots,Y_n)$ for the polynomial obtained from the original ${}^hf_{i+1}(X_0,\ldots,X_n)$ by means of the change of variables from $X_0,\ldots,X_n$ to $Y_0,\ldots,Y_n$). Observe that $\rho_i^k M_{{}^hf_{i+1}}\in k[Y_0,\ldots,Y_{n-i}]^{\delta_i\times\delta_i}$ holds and that the entries are homogeneous polynomials. Therefore the entries of the matrix $\rho_i^k M_{{}^hf_{i+1}}$ can be computed by an ordinary straight–line program in $k[Y_0,\ldots,Y_{n-i}]$ of length $L\delta_i^{O(1)}$ and depth $O(log_2\delta_i)+\ell$ from the coefficients of q_i, $v_{n-i+1}^{(i)},\ldots,v_n^{(i)}$ and from ρ_i. This is simply done by applying the by the input given evaluation scheme of ${}^hf_{i+1}$ and plugging in the entries of the matrices $v_{n-i+1}^{(i)}(M_i),\ldots,v_n^{(i)}(M_i)$.
Next, we use Berkowitz' well parallelizable and division free algorithm [4] in order to compute the (homogeneous) coefficients of the characteristic polynomial $\chi_{\rho_i^k{}^hf_{i+1}}\in k[Y_0,\ldots,Y_{n-i},T]$ of the homothety given modulo J_i by $\rho_i^k{}^hf_{i+1}$. The computation of the coefficients of $\chi_{\rho_i^k{}^hf_{i+1}}$ from the entries of $\rho_i^k M_{{}^hf_{i+1}}$ requires an additional free and well parallelizable straight–line program in $k[Y_0,\ldots,Y_{n-i}]$ of size $\delta_i^{O(1)}$.
Now we apply Lemma 6 in order to obtain the (homogeneous) coefficients with respect to T of a square free representation $\chi^*_{\rho_i^k{}^hf_{i+1}}\in k[Y_0,\ldots,Y_{n-i},T]$ of $\chi_{\rho_i^k{}^hf_{i+1}}$. This squarefree representation is the minimal polynomial $m_{\rho_i^k{}^hf_{i+1}}$ multiplied by some non–zero homogeneous $\theta\in k[Y_0,\ldots,Y_{n-i}]$. Observe that the coefficients of $m_{\rho_i^k{}^hf_{i+1}}$ are homogeneous too. The computation of θ and the remaining coefficients of $\chi^*_{\rho_i^k{}^hf_{i+1}} = \theta\cdot m_{\rho_i^k{}^hf_{i+1}}$ requires one more division free and well parallelizable straight–line program in $k[Y_0,\ldots,Y_{n-i}]$ of size $\delta_i^{O(1)}$.
Putting everything together we obtain a division free straight–line program with FOR gates in $k[Y_0,\ldots,Y_{n-i}]$ of size $L\delta_i^{O(1)}+\Lambda_i$ and depth $O(log_2\delta_i)+\ell+\lambda_i$ which computes the homogeneous constant coefficient, say $\mathcal{A}\in k[Y_0,\ldots,Y_{n-i}]$ of $\theta\cdot m_{\rho_i^k{}^hf_{i+1}}$ and the polynomial θ. We remark that $\mathcal{A}$ has degree at most $(d\delta_i)^{O(1)}$. By Lemma 11 this coefficient has the form $\mathcal{A}=\theta\rho_i^m a_0$ for some $m\leq d\delta_i$ and is therefore non–zero. Since $m_{{}^hf_{i+1}}$ annihilates ${}^hf_{i+1}$ modulo J_i we conclude that $a_0\in({}^hf_{i+1},J_i)$ holds in $k[Y_0,\ldots,Y_n]=k[X_0,\ldots,X_n]$. Since the polynomial $\mathcal{A}$ can be evaluated by a division free straight–line program with FOR gates in $k[Y_0,\ldots,Y_{n-i}]$ of size $L(d\delta_i)^{O(1)}+\Lambda_i$ and depth $O(log_2 d\delta_i)+\ell+\lambda_i$, there exists by Lemma 8 a correct test sequence of length $(L\delta_i^{O(1)}+\Lambda_i)^2$ in k^{n-i+1} for this complexity class. Because the polynomial $\mathcal{A}$ is non–zero we find in sequential time $(L(d\delta_i)^{O(1)}+\Lambda_i)^3$ and parallel time $O(log_2 d\delta_i)+\ell+\lambda_i$

a point $\gamma = (\gamma_0, \ldots, \gamma_{n-i}) \in k^{n-i+1}$ in this correct test sequence such that $\mathcal{A}(\gamma) \neq 0$ holds. Since $\mathcal{A}$ is homogeneous we may construct in the obvious way from the coordinates $\gamma_0, \ldots, \gamma_{n-i}$ of γ a linear change of the variables $Y_0, \ldots, Y_n$ into new ones, say $Y_0', \ldots, Y_n'$, such that the homogeneous polynomial $\mathcal{A} \in k[Y_0', \ldots, Y_{n-i}'] = k[Y_0, \ldots, Y_{n-i}]$ given by the equation $\mathcal{A}'(Y_0, \ldots, Y_{n-i}) = \mathcal{A}(Y_0, \ldots, Y_{n-i})$ becomes monic in Y_{n-i}'. This implies that the homogeneous polynomial $A_0(Y_0', \ldots, Y_{n-i}') = a_0(Y_0, \ldots, Y_{n-i})$ is monic too in Y_{n-i}'.

Therefore, if we replace the variables $Y_0, \ldots, Y_n$ by the new variables $Y_0', \ldots, Y_{n-i}', Y_{n-i+1}' := Y_{n-i+1}, \ldots, Y_n' := Y_n$ we obtain a Noether normalization with respect to the ideal $\mathcal{I}_{i+1} = ({}^h f_{i+1}, J_i)$ and hence with respect to J_{i+1}.

In the next part of our algorithm we shall make use of the eliminating form a_0 instead of the "accidental" polynomial $\mathcal{A}$ which comes from our specific algorithm. We consider a_0, $\mathcal{A}$, θ and ρ_i as forms in the variables $Y_0', \ldots, Y_{n-i}'$ which are related by the identity

$$a_0 = \frac{\mathcal{A}}{\theta \rho_i^k}.$$

Let $\mu := \theta \rho_i^k$ and observe that μ is monic in Y_{n-i}'. The exponent $k \leq d$ is known and the forms $\mathcal{A}$, θ and ρ_i (and hence $\mu = \theta \rho_i^k$) are represented in our algorithm by a *division free* straight–line program Σ with FOR gates in $k[Y_0', \ldots, Y_{n-i}'] = k[Y_0, \ldots, Y_{n-i}]$ which has size $L(d\delta_i)^{O(1)}$ and depth $O(log_2 d\delta_i)) + \ell + \lambda_i$. The degrees of μ and $\mathcal{A}$ are of order $(d\delta_i)^{O(1)}$.

Using just two FOR gates of cost $(d\delta_i)^{O(1)}$ we interpolate $\mathcal{A}$ and μ with respect to the variable Y_{n-i}' in $(d\delta_i)^{O(1)}$ (at least $max\{deg_{Y_{n-i}'}\mu, deg_{Y_{n-i}'}\mathcal{A}\}+1$) many distinct points of k and compute their coefficients which are polynomials belonging to $k[Y_0', \ldots, Y_{n-i-1}']$. Since the square matrix with entries in $k[Y_0', \ldots, Y_{n-i}']$ which correspond to the division (with remainder) of $\mathcal{A}$ by μ in the principal ideal domain $k(Y_0', \ldots, Y_{n-i-1}')[Y_{n-i}']$ is unimodular and since the identity $a_0 = \frac{\mathcal{A}}{\mu}$ holds, we are able to compute the coefficients of the polynomial a_0 with respect to Y_{n-i}' (and hence a_0 itself) from the circuit Σ in sequential and parallel time $(d\delta_i)^{O(1)}$ and $O(log_2 d\delta_i)$ respectively. Therefore we may suppose that a_0 (and its coefficients with respect to Y_{n-i}') is given by a *division free* straight–line program with FOR gates in $k[Y_0', \ldots, Y_{n-i}'] = k[Y_0, \ldots, Y_{n-i}]$ of size and depth $L(d\delta_i)^{O(1)} + \Lambda_i$ and $O(log_2 d\delta_i) + \ell + \lambda_i$ respectively.

Recursive Generation of a Primitive Element. The next recursive step is the computation of a primitive element u_{i+1} modulo J_{i+1}. We assume without loss of generality that the variables $Y_0, \ldots, Y_n$ are already in Noether position with respect to the ideal $\mathcal{I}_{i+1} = ({}^h f_{i+1}, J_i)$ (see "recursive Noether normalization"). So, we have the following integral ring extension

$$R := k[Y_0, \ldots, Y_{n-i-1}] \hookrightarrow k[Y_0, \ldots, Y_n]/({}^h f_{i+1}, J_i) = k[Y_0, \ldots, Y_n]/(\mathcal{I}_{i+1})$$

If we denote the fraction field of R by K (i.e. $K := k(Y_0, \ldots, Y_{n-i-1})$) it is clear that $\mathcal{I}_{i+1}$ generates a zero dimensional ideal in $K[Y_{n-i}, \ldots, Y_n]$. We are going to apply the method described in Subsection 3.3. For this purpose we introduce new variables, say $T_{n-i}, \ldots, T_n$, and with respect to them for $n-i \le j \le n$ the rings and fields $R_j := R[T_{n-i}, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n]$ and $K_j := K(T_{n-i}, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n)$ introduced in 3.3. For $n-i \le j \le n$ we are going to construct polynomials $h_j \in R[T]$ and $g_j \in R_j[T]$ with the following properties :

- $h_j \in R[T]$ is monic in T of degree at most $d\delta_i$ in T and the polynomial $h_j(Y_j)$ belongs to the ideal $\mathcal{I}_{i+1}$
- $g_j \in R_j[T]$ is monic in T and has total degree at most $d\delta_i$. The polynomial $g_j(Z_j)$ belongs to the ideal $K_j \bigotimes_R \mathcal{I}_{i+1}$

Our aim is to represent the coefficients of h_j and g_j with respect to T by an evaluation scheme. In order to do that we need FOR loops.

With the notations and results of "recursive Noether normalization" there is already given an equation $a_0 \in R[Y_{n-i}] = k[Y_0, \ldots, Y_{n-i}]$ of integral dependence of Y_{n-i} over R modulo the ideal $\mathcal{I}_{i+1}$. This polynomial a_0 is given by a straight–line program with FOR gates in $k[Y_0, \ldots, Y_{n-i}]$ of length $L\delta_i^{O(1)} + \Lambda_i$ and depth $O(log_2\delta_i) + \ell + \lambda_i$ and has degree $d\delta_i$. Moreover it is homogeneous with respect to the grading of $R[Y_{n-i}]$.

Fix $n-i \le j \le n$ and consider the variable Y_j and the linear form $Z_j := T_{n-i}Y_{n-i} + \cdots + T_{j-1}Y_{j-1} + T_{j+1}Y_{j+1} + \cdots + T_nY_n \in R_j[Y_{n-i}, \ldots, Y_n]$ in the variables $Y_{n-i}, \ldots, Y_n$ (see Subsection 3.3). To the homotheties given by these forms in the δ_i–dimensional algebras $K[Y_{n-i}, \ldots, Y_n]/(J_i)$ and $K_j[Y_{n-i}, \ldots, Y_n]/(J_i)$ and to their bases given by the forms $1, u_i, \ldots, u_i^{\delta_i - 1}$ correspond matrices $M_{Y_j} \in K^{\delta_i \times \delta_i}$ and $M_{Z_j} \in K_j^{\delta_i \times \delta_i}$. The characteristic polynomials of these matrices induce polynomials in the variables Y_{n-i}, Y_j and Y_{n-i}, Z_j respectively which have degree at most δ_i in these indeterminates. Since the variables $Y_0, \ldots, Y_n$ are in Noether position with respect to J_i the coefficients of these polynomials belong to R and R_j respectively. These coefficients are homogeneous elements of degree at most δ_i of their respective graded rings. Let us denote the polynomials introduced in this way as $H_j \in R[Y_{n-i}, Y_j]$ and $G_j \in R_j[Y_{n-i}, Z_j]$. They have the following properties :

- H_j is monic in Y_j and belongs to the ideal J_i
- G_j is monic in Z_j and belongs to the ideal $K_j \otimes_R J_i$.

These polynomials can be computed from the results of the circuit Γ_i as in "recursive Noether normalization" by means of the division free and well parallelizable algorithm [4] in sequential time $\delta_i^{O(1)}$. This means that their coefficients with respect to the variables Y_j and Z_j are given by division free straight–line programs with FOR gates in the ring $R[Y_{n-i}] = k[Y_0, \ldots, Y_{n-i}]$ and $R_j[Y_{n-i}] = k[T_{n-i}, \ldots, T_{j-1}, T_{j+1}, \ldots, T_n, Y_{n-i}]$ and that these straight–line programs have length $\delta_i^{O(1)} + \Lambda_i$ and depth $O(log_2\delta_i) + \lambda_i$. (Without going into details we remark here that one has to use the same type of subroutine to eliminate divisions by a monic polynomial in Y_{n-i} as we did at the end of "recursive Noether normalization" when computing a_0).

For $n-i \leq j \leq n$ we are now going to construct the announced polynomials $h_j \in R[T]$ and $g_j \in R_j[T]$. For this purpose we need the coefficients with respect to Y_{n-i} of a_0 and the coefficients with respect to Y_{n-i}, Y_j and Y_{n-i}, Z_j of H_j and G_j. This requires interpolation of these polynomials in $d\delta_i$ points of k which we have to substitute for the variable Y_{n-i} (recall that H_j and G_j are already given by their coefficients with respect to Y_j and Z_j respectively). For this interpolation we need $2i+1$ parallel organized FOR gates which originate a total sequential time cost of $(2i+1)d\delta_i$.

Thus the coefficients with respect to the variable Y_{n-i} of a_0 and for $n-i \leq j \leq n$ the coefficients with respect to Y_{n-i}, Y_j and Y_{n-i}, Z_j of the polynomials H_j and G_j are represented by a division free straight–line program with FOR gates in $R = k[Y_0, \ldots, Y_{n-i-1}]$ which has length $(Lid\delta_i)^{O(1)} + \Lambda_i$ and depth $O(log_2(d\delta_i) + \ell) + \lambda_i$. Once these coefficients in R are given we are able to compute effectively in the graded k–algebras which for $n-i \leq j \leq n$ are defined as follows :

- $B_j := R[Y_{n-i}, Y_j]/(a_0, H_j)$.
- $C_j := R_j[Y_{n-i}, Z_j]/(a_0, G_j)$.

Observe that B_j is a free R–module with basis represented by the set of monomials

$$\{Y_{n-i}^{\alpha} Y_j^{\beta} \ : 0 \leq \alpha < deg\ (a_0), 0 \leq \beta < deg\ (H_j)\}$$

and that C_j is a free R_j–module with basis represented by the set of monomials

$$\{Y_{n-i}^{\alpha} Z_j^{\beta} \ : 0 \leq \alpha < deg\ (a_0), 0 \leq \beta < deg\ (G_j)\}.$$

The ranks of the free modules B_j and C_j are therefore bounded by $d\delta_i^2$. Since the coefficients of a_0 with respect to Y_{n-i} and the coefficients of H_j with respect to Y_{n-i}, Y_j are given, we know effectively (with no extra cost) the multiplication tensor of B_j. In the same sense the multiplication tensor of C_j is available. From this data we compute the characteristic polynomial $\chi_{Y_j} \in R[T]$ of the homothety in B_j given by the multiplication by Y_j and the characteristic polynomial $\chi_{Z_j} \in R_j[T]$ of the homothety in C_j given by Z_j. We put $h_j := \chi_{Y_j}$ and $g_j := \chi_{Z_j}$. One verifies immediately that h_j and g_j have the required properties (for this purpose observe that $a_0 \in \mathcal{I}_{i+1}$, $H_j \in J_i \subseteq \mathcal{I}_{i+1}$ and $G_j \in K_j \otimes_R J_i \subseteq K_j \otimes_R \mathcal{I}_{i+1}$ holds).

Applying Lemma 9 we obtain a linear form u_{i+1} which is a primitive element for $K[Y_{n-i}, \ldots, Y_n]/(\mathcal{I}_{i+1})$, a homogeneous squarefree polynomial $q(u_{i+1}) \in \sqrt{\mathcal{I}_{i+1}}$ and homogeneous polynomials $\rho_{i+1} \in R$ and $v_{n-i}, \ldots, v_n \in R[T]$ such that $\rho_{i+1} Y_j - v_j(u_{i+1}) \in \sqrt{\mathcal{I}_{i+1}}$ holds for $n-i \leq j \leq n$. Observe that $q(u_{i+1})$ is monic in u_{i+1}.

These items (and their homogeneous coefficients in $R = k[Y_0, \ldots, Y_{n-i-1}]$) are represented by a division free straight– line program with FOR gates in $k[Y_0, \ldots, Y_{n-i-1}]$ whose length and depth are $(Lid\delta_i)^{O(1)} + \Lambda_i$ and $O(log_2(d\delta_i) + \ell) + \lambda_i$ respectively.

Clearing Extraneous Irreducible Components. In the subsection "recursive Noether normalization" we obtained a fairly explicit description of the projective variety defined by the ideal $\mathcal{I}_{i+1}$ and of the localization of the ideal $\mathcal{I}_{i+1}$ in $K[Y_{n-i}\ldots,Y_n]$. However this variety may contain irreducible components at infinity (extraneous components) or components which count in $\mathcal{I}_{i+1}$ (or its localization) with higher multiplicities.

The situation in the toric case is the same with "extraneous components at infinity" replaced by "extraneous components contained in the union of hyperplanes $V(\prod_{i=1}^{n} X_i)$". We show now how extraneous components and multiplicities which appear in a natural way when cutting by the hypersurfaces $V({}^hf_1),\ldots,V({}^hf_n)$, can be cleared during the process. Let us concentrate upon the affine case.

First recall that J_{i+1} is the intersection of all those homogeneous prime ideals of codimension $i+1$ which contain ${}^hf_{i+1}$ and J_i but do not contain X_0. We observe also that the (homogeneous) primes of codimension $i+1$ of $R[Y_{n-i},\ldots,Y_n]=k[Y_0,\ldots,Y_n]$ which contain the ideal $({}^hf_1,\ldots,{}^hf_{i+1})$ but not X_0 correspond to the primes in the localized ring $K[Y_{n-i},\ldots,Y_n]$ which contain the ideal generated by $\mathcal{I}_{i+1}$ but not X_0. We are going now to determine the irreducible components at finite distance of the projective variety defined by ${}^hf_1,\ldots,{}^hf_{i+1}$ (or else by $\mathcal{I}_{i+1}$) doing computations in $K[u_{i+1}]$. These components characterize geometrically the ideal J_{i+1}. For $1\le j\le i+1$ let us write $F_j := {}^h f_j(Y_0,\ldots,Y_{n-i-1},\frac{v_{n-i}(u_{i+1})}{\rho_{i+1}},\ldots,\frac{v_n(u_{i+1})}{\rho_{i+1}})\in K[u_{i+1}]$. Let $h\in R[u_{i+1}]$ be a greatest common divisor of the polynomials $F_1(u_{i+1}),\ldots,F_{i+1}(u_{i+1})$ and $q(u_{i+1})$ with respect to the principal domain $K[u_{i+1}]$.

Since $q(u_{i+1})$ is monic in u_{i+1} we may suppose without loss of generality that h is monic too in u_{i+1}. Moreover we may suppose that h is homogeneous with respect to the grading of $R[u_{i+1}]$. The coefficients of h with respect to the variable u_{i+1} can be computed from the coefficients of $F_1,\ldots,F_{i+1}$, $q\in K[u_{i+1}]$ by a well parallelizable and division free straight–line program of size $(d\delta_i)^{O(1)}$. This means that these coefficients can be computed by a division free straight–line program with FOR gates in $R=k[Y_0,\ldots,Y_{n-i}]$ of size $(Lid\delta_i)^{O(1)}+\Lambda_i$ and depth $O(log_2(d\delta_i)+\ell)+\lambda_i$ (here again we apply the same trick as at the end of "recursive Noether normalization" in order to eliminate division by ρ_{i+1}).

Observe now that the maximal ideals in $K[Y_{n-i},\ldots,Y_n]$ that contain the polynomial $h(u_{i+1})$ and for $n-i\le j\le n$ the polynomials $\rho_{i+1}Y_j-v_j(u_{i+1})$, are in one to one correspondence with the codimension $i+1$ prime ideals in $R[Y_{n-i},\ldots,Y_n]$ which contain $({}^hf_1,\ldots,{}^hf_{i+1})$. Moreover $h(u_{i+1})$ is squarefree since it is a divisor of the form $q_{i+1}(u_{i+1})$ which is monic in u_{i+1}.

Note that $X_0\in k[X_0,\ldots,X_n]=k[Y_0,\ldots,Y_n]$ is a linear form in $Y_0,\ldots,Y_n$. Substituting in this linear form for the variables $Y_{n-i},\ldots,Y_n$ the polynomials $\frac{v_{n-i}(u_{i+1})}{\rho_{i+1}},\ldots,\ \frac{v_n(u_{i+1})}{\rho_{i+1}}\in K[u_{i+1}]$ and clearing the denominators ρ_{i+1} we obtain a representation in $R[u_{i+1}]$ of the residue class of $\rho_{i+1}X_0$ modulo the ideal generated by $\mathcal{I}_{i+1}$ in $K[Y_{n-i},\ldots,Y_n]$. Let $G\in R[u_{i+1}]$ be this represen-

tation. Now we compute the greatest common divisor of $G(u_{i+1})$ and $h(u_{i+1})$ in the principal ideal domain $K[u_{i+1}]$. This greatest common divisor is represented by a squarefree polynomial $h_1 \in R[u_{i+1}]$ which divides $h \in R[u_{i+1}]$. Observe that h_1 is monic in u_{i+1} and homogeneous with respect to the grading of $R[u_{i+1}]$. The maximal ideals of $K[Y_{n-i}, \ldots, Y_n]$ which contain $h_1(u_{i+1})$ and for $n-i \leq j \leq n$ the polynomials $\rho_{i+1}Y_j - v_j(u_{i+1})$, are in one to one correspondence with the codimension $i+1$ prime ideals of $R[Y_{n-i}, \ldots, Y_n]$ which contain $(X_0, {}^h f_{i+1}, \ldots, {}^h f_n)$.
Now it is easy to see that the maximal ideals in $K[Y_{n-i}, \ldots, Y_n]$ which contain for $n-i \leq j \leq n$ the polynomials $\rho_{i+1}Y_j - v_j(u_{i+1})$ and the homogeneous and monic in u_{i+1} polynomial

$$q_{i+1}(u_{i+1}) := \frac{h(u_{i+1})}{h_1(u_{i+1})}$$

correspond exactly to the codimension $i+1$ prime ideals which contain the radical ideal J_{i+1} (these are also the associated primes of J_{i+1}). Finally the polynomials $v_{n-i}^{(i+1)}(u_{i+1}), \ldots, v_n^{(i+1)}(u_{i+1})$ are defined (and computed) as the remainders of the division in $R[u_{i+1}]$ of $v_{n-i}(u_{i+1}) \ldots, v_n(u_{i+1})$ by the monic in u_{i+1} polynomial $q_{i+1}(u_{i+1})$ (note that this leaves the ideal in question unchanged). In this way we obtain an explicit description of an isomorphism $K[Y_{n-i}, \ldots, Y_n]/J_{i+1} \cong K[u_{i+1}]/(q_{i+1}(u_{i+1}))$ as wanted. Since h_1 is monic in u_{i+1} we are able to compute by the same trick as before q_{i+1} by a *division free* straight–line program. Because q_{i+1} is again monic in u_{i+1} the same argument applies to $v_{n-i}^{(i+1)}, \ldots, v_n^{(i+1)}$. Finally we observe that ρ_{i+1} and the coefficients with respect to the variable u_{i+1} of $q(u_{i+1})$ and $v_{n-i}^{(i+1)}(u_{i+1}), \ldots, v_n^{(i+1)}(u_{i+1})$ can be computed by a division free straigth–line program with FOR gates of size $(id\delta_i L)^{O(1)} + \Lambda_i$ and depth $O(log_2(id\delta_i) + \ell) + \lambda_i$. We conclude this subsection with the remark that the toric case can be treated in exactly the same way replacing at the end of the construction for the affine case the linear form X_0 by the homogeneous polynomial $\prod_{i=0}^{n} X_i$.

4.2 Proof of Theorems 1 and 2

End of the Proof of Theorem 1. Iterating $n-1$ times the recursive procedure described in the last subsection we obtain finally a division free arithmetic network with FOR gates in $k[X_1, \ldots, X_n]$ of size $(nd\delta L)^{O(1)}$ and depth $O(n(log_2(\delta d) + \ell))$ which represents the following items :

– A linear change of variables $X_0, \ldots, X_n$ into new variables $Y_0, \ldots, Y_n$
– a linear form u_n in $k[Y_0, Y_1, \ldots, Y_n]$
– non–zero forms q_n, $v_1^{(n)}, \ldots, v_n^{(n)} \in k[Y_0, T]$.

These items have the following properties :
the canonical isomorphism $k[Y_0] \longrightarrow k[Y_0, \ldots, Y_n]/J_n$ is an integral ring extension and we have an isomorphism $k(Y_0)[Y_1, \ldots, Y_n]/J_n \cong k(Y_0)[T]/(q_n)$. The

form q_n is monic in T and we have $max\{deg_T v_1^{(n)}, \ldots, deg_T v_n^{(n)}\} < deg q_n = deg_T q_n = \delta_n$. Moreover the total degrees of $v_1^{(n)}, \ldots, v_n^{(n)}$ are bounded by a polynomial in $\delta_n = deg V(J_n)$. Up to now we did not consider the non–zero linear form $H \in k[X_0, \ldots, X_n]$ given by the input. From the data q_n , $v_1^{(n)}, \ldots, v_n^{(n)}$ we compute in sequential time $(nd\delta_n)^{O(1)}$ by a well parallelizable division free straigth–line program in $k[Y_0, T]$ a non–zero form $g \in k[Y_0, T]$ of degree at most $deg V(J_n) = \delta_n$ such that $g(Y_0, H) \in J_n$ holds. Moreover g is monic in T . Similarly we compute a non–zero form $q \in k[Y_0, T]$ monic in T of degree at most δ_n such that $q(Y_0, X_0) \in J_n$ holds. Replacing Y_0 by a generic linear combination of Y_0 and T we may assume that q is also monic in Y_0 . In particular we have $q(Y_0, 1) \neq 0$.

Since $g \in k[Y_0, T]$ is monic in T , the greatest common divisor of g and $q(Y_0, 1)$ in the principal ideal domain $k(T)[Y_0]$ is one. Therefore the resultant of g and $q(Y_0, 1)$ with respect to Y_0 is a non–zero polynomial, say $p \in k[T]$. From the fact that $g(Y_0, H)$, $q(Y_0, X_0)$ belong to the ideal J_n we conclude that $p(H)$ vanishes on the zero–dimensional affine variety $V(I_n) = V(f_1, \ldots, f_n)$. The non–zero polynomial $p(T)$ is the final output of our algorithm in the affine case. Note that the size of the (square) matrices which are involved in this part of our proof never exceeds δ and that the computations we have to perform in order to obtain the polynomial $p \in k[T]$ from the items of the n –th step of our algorithm (essentially greatest common divisor and resultant computations) can be realized by an additional well parallelizable and division free arithmetic network of size $\delta^{O(1)}$. □

The end of the proof of Theorem 2 is similar.

Acknowledgments : We thank the unknown referees for their suggestions and critical remarks on a first draft of this paper.

References

1. M.E. Alonso, E. Becker, M.-F. Roy, T. Wörmann : Zeroes, multiplicities and idempotents for zerodimensional systems, to appear in *Proc. MEGA'94*, Birkhäuser Progress in Mathematics.
2. C. Berenstein and A. Yger : Effective Bézout identities in $\mathbb{Q}[X_1, \ldots, X_n]$, *Acta Math.,* **166** (1991) 69-120
3. C. Berenstein and A. Yger : Une Formule de Jacobi et ses conséquences, *Ann Sci. E.N.S.,* 4^{ieme} *série,* **24** (1991) 363-377
4. S.J. Berkowitz : On computing the determinant in small parallel time using a small number of processors, *Inf. Proc. Letters* **18** (1984) 147–150.
5. J. Briançon : Sur le degré des relations entre polynômes, *Comptes Rendues de l'Acad. des Sci. Paris, t.,* Série I Math (1982) 553-556.
6. D. W. Brownawell : Bounds for the degree in the Nullstellensatz, *Annals of Math.* **126** (1987) 577-591.

7. L.Caniglia, A.Galligo and J.Heintz : Some new effectivity bounds in computational geometry, In *Proc. AAECC–6, T. Mora (Ed.), Springer LN in Comp. Sci.* **357**(1989) 131–152.
8. L. Caniglia, A. Galligo and J. Heintz : Borne simplement exponentielle pour les degrés dans le théorème des zéros sur un corps de caractéristique quelconque, *C.R. Acad. Sci. Paris, Série I, Math.* **307** (1988) 255-258.
9. J. Canny : Some algebraic and geometric problems in PSPACE, *Proc. 20-th Ann. ACM Symp. Theory of Computing* (1988) 460-467.
10. A.L. Chistov : Polynomial –time computation of the dimension of components of algebraic varieties in zero–characteristic, *Preprint Université Paris Val de Marne.*
11. A.L. Chistov, D.Yu. Grigor'ev : Subexponential time solving systems of algebraic equations, LOMI Preprints E-9-83, E-10-83, Leningrad (1983).
12. F. Cucker, M. Karpinski, P. Koiran, T. Lickteig, K. Werther : On real Turing machines that toss coins.
13. R.A. De Millo, R.J. Lipton : A probabilistic remark on algebraic program testing, *Inf. Proc. Letters*, vol. **7**, No. 4 (1978) 193–195.
14. A. Dickenstein , N. Fitchas, M. Giusti, C. Sessa : The membership problem of unmixed ideals is solvable in single exponential time, *Discrete Applied Mathematics* **33** (1991) 73–94.
15. T.W. Dubé : A combinatorial proof of Effective Nullstellensatz, *J. Symb. Comp.* **15** (1993) 277–296.
16. D. Duval : Évaluation Dynamique et clôture algébrique en Axiom, *Journal of Pure and Applied Algebra* (1994).
17. N. Fitchas, M. Giusti and F. Smietanski : Sur la complexité du théorème des zéros, to appear in *Proceedings of the Second International Conference on Approximation and Optimization*, La Habana, 1993, J. Gudatt, ed., Peter Lang Verlag, Frankfurt (1995).
18. W. Fulton : Intersection theory, Ergebnisse der Mathematik, 3 Folge Band 2, Springer Verlag (1984).
19. C.B. García and W. I. Zangwill : Pathways to solutions, fixed points and equilibria, *Prentice-Hall, N.J.* (1981)
20. J.von zur Gathen : Parallel arithmetic computations : a survey, *Proc. 13-th Conf. MFCS.* Springer LN Comp. Sci. **233**(1986) 93–112.
21. M.Giusti and J.Heintz : La détermination des points isolés et de la dimension d'une variété algébrique peut se faire en temps polynomial, In *Computational Algebraic Geometry and Commutative Algebra*, Proceedings of the Cortona Conference on Computational Algebraic Geometry and Commutative Algebra, D. Eisenbud and L. Robbiano, eds., Symposia Matematica, vol. XXXIV, Istituto Nazionale di Alta Matematica, Cambridge University Press (1993).
22. M. Giusti, J. Heintz and J. Sabia : On the efficiency of effective Nullstellensätze, *Computational Complexity* **3** (1993) 56–95.
23. L. González Vega, G. Trujillo : Using symmetric functions to describe the solution of a zero–dimensional ideal, *in this volume.*
24. K. Haegele, J.L. Montaña : Polynomial random test for the equivalence problem of integers given by arithmetic circuits. *Stasiakte Humboldt–Universität. Preprint Series X901–B*
25. J. Heintz : Fast quantifier elimination over algebraically closed fields, *Theoret. Comp. Sci.* **24** (1983) 239-277.
26. J. Heintz : On the computational complexity of polynomials and bilinear mappings, in *Proc. 5th International Conference Applied Algebra, Algebraic Algorithms and*

Error–Correcting Codes AAECC–5, Menorca 1987, L. Huguet and A. Poli, eds., Springer LN Comput. Sci. **356** (1989) 269–300.
27. J. Heintz and J. Morgenstern : On the intrinsic complexity of elimination theory, *J. of Complexity* **9** (1993) 471–498
28. J. Heintz and C.P. Schnorr : Testing Polynomials which are easy to compute, *Proc. 12th Ann. ACM Symp. on Computing* (1980) 262–268; also in *Logic and Algorithmic. An International Symposium held in Honour of Ernst Specker*, Monographie No. **30** de l'Enseignement de Mathématiques, Genève (1982) 237–254.
29. J. Heintz and M. Sieveking : Absolute primality of polynomials is decidable in random polynomial time in the number of the variables, *8th International Colloquium on Automata, Languages and Programming ICALP 81*, Springer LN Comput. Sci. **115** (1981) 16–28.
30. O. Ibarra, H. Moran : Probabilistic algorithms for deciding equivalence of straight–line programs, *J. ACM* **30**, Vol. 1, 217–228
31. E. Kaltofen : Greatest common divisors of polynomials given by straight line programs, *J. ACM* **35**, No. 1 (1988) 234–264.
32. E. Kaltofen : Factorization of polynomials given by straight–line programs, *Randomness in Computation*, Advances in Computing Research **5**, S. Micali, ed., JAI Press Inc., Greenwich, CT. (1989) 375–412.
33. J. Kollár : Sharp Effective Nullstllensatz, *J. of A.M.S.* **1** (1988) 963-975.
34. T.Krick and L.M.Pardo : A Computational Method for Diophantine Approximation, To appear in *Proc. MEGA'94*, Birkhäuser Progress in Mathematics.
35. T. Krick, L.M. Pardo : Une approche informatique pour l'approximation diophantienne, *Comptes Rendues de l'Acad. des Sci. Paris, t.* **318**, Série I, no. 5 , (1994) 407–412.
36. Y.N. Lakshman, D. Lazard : On the complexity of zero–dimensional algebraic systems, *Effective methods in algebraic geometry (Castiglioncello, 1990), Prog. Math.*, vol. **94**, Birkhäuser Boston (1991) 217–225.
37. D. Lazard : Résolution des systèmes d'équations algébriques, *Theor. Comp. Sci.* **15** (1981) 77-110.
38. F.S. Macaulay : The Algebraic Theory of Modular Systems, *Cambridge University Press* (1916)
39. M. Moeller : Systems of algebraic equations solved by means of endomorphisms, in *Proc. 10th International Conference Applied Algebra, Algebraic Algorithms and Error–Correcting Codes AAECC–10*, Puerto Rico 1993, G. Cohen, T. Mora and O. Moreno, eds., Springer LN Comput. Sci.
40. J.L. Montaña and L.M. Pardo : Lower Bounds for Arithmetic Networks, *AAECC*, vol. **4** (1993), 1–24.
41. L.M. Pardo : How lower and upper complexity bounds meet in elimination theory, *in this volume.*
42. P. Philippon : Dénominateurs dans le théorème des zéros de Hilbert, *Acta. Arith.* **58** (1991) 1–25.
43. J. Sabia and P. Solernó : Bounds for traces in complete intersections and degrees in the Nullstellensatz, *To appear in AAECC Journal* (1993).
44. J.T. Schwartz : Fast probabilistic algorithms for verification of polynomial identities, *J. ACM* **27** (1980) 701–717
45. M. Shub, S. Smale : Complexity of Bézout's theorem V : Polynomial time, *Theoretical Comp. Sci.* **133** (1994) 141–164.
46. M. Shub, S. Smale : On the intractability of Hilbert's Nullstellensatz and an algebraic version of $NP \neq P?$, *IBM Research Report*, Yorktown Heights (1994)

47. V. Strassen : Algebraic Complexity Theory, *Handbook of Theoretical Computer Science*, ch. **11** (1990) 634–671.
48. V. Strassen : Vermeidung von Divisionen, *Crelle J. Reine Angew. Math.* **264** (1973) 184-202.
49. R.E. Zippel : Probabilistic Algorithms for Sparse Polynomials, *LNCS*, Vol **72** (1979) 216–226