The Complexity of Factors of Multivariate Polynomials[*]

Peter Bürgisser
Dept. of Mathematics and Computer Science
University of Paderborn
D-33095 Paderborn, Germany
buergisser@upb.de

Abstract

The existence of string functions, which are not polynomial time computable, but whose graph is checkable in polynomial time, is a basic assumption in cryptography. We prove that in the framework of algebraic complexity, there are no such families of polynomial functions of p-bounded degree over fields of characteristic zero. The proof relies on a polynomial upper bound on the approximative complexity of a factor g of a polynomial f in terms of the (approximative) complexity of f and the degree of the factor g. This extends a result by Kaltofen (STOC 1986). The concept of approximative complexity allows to cope with the case that a factor has an exponential multiplicity, by using a perturbation argument. Our result extends to randomized (two-sided error) decision complexity.

1 Introduction

Checking or verifying a solution to a computational problem might be easier than computing a solution. In a certain sense, this is the contents of the famous $P \neq NP$ hypothesis. In [30] Valiant made an attempt to clarify the principal relationship between the complexity of checking and evaluating. In particular, he asked whether any (string) function, that can be checked in polynomial time, can also be evaluated in polynomial time. Cryptographers hope that the answer to this question is negative, since it turns out to be intimately connected to the existence of one-way functions. Indeed, the inverse φ of a one-way function is not polynomial time computable, but membership to the graph of φ can be decided in polynomial time. The converse is also known to be true [18, 26] and equivalent to $P \neq UP$.

The goal of this paper is to investigate the relationship between the complexity of computational and decisional tasks in an algebraic framework of computation. Through-

out the paper, k denotes a fixed field of characteristic zero. Are there families of polynomials (φ_n) over k, for which checking can be done with a polynomial number of arithmetic operations and tests, but which cannot be evaluated with a polynomial number of arithmetic operations? If we require that (φ_n) is a p-family, i.e., the number of variables and the degree of φ_n grow at most polynomially in n, then our main result states that the answer to this question is no! Actually, the result is slightly weaker and can be stated as follows (the statement with detailed bounds is in Corollary 5.1).

Theorem 1.1 *The approximative complexity $\underline{L}(\varphi)$ of a polynomial φ is polynomially bounded in the decision complexity of the graph of φ and the degree of φ. This remains true if we allow randomization with two-sided error.*

Hereby, the decision complexity of the graph of φ measures the number of arithmetic operations and equality-tests in the model of algebraic computation trees. If k is ordered, for instance $k = \mathbb{R}$, we allow also $\leq$-tests. (For formal definitions see [9].) The approximative complexity $\underline{L}(\varphi)$ measures the minimal cost of "approximative straight-line programs" computing approximations of φ with any precision required, for a formal definition see Section 3.

It is not known whether the degree restriction in Theorem 1.1 can be omitted. However, in view of the discussion following Lemma 1.2 below, this seems unlikely.

Sturtivant and Zhang [29] obtained the following related result, which excludes the existence of certain one-way functions in the algebraic framework of computation. Let $\psi: k^n \to k^n$ be bijective such that ψ as well as ψ^{-1} are polynomial mappings. Then the complexity to evaluate ψ is polynomially bounded in the complexity to evaluate the inverse ψ^{-1} and the maximal degree of the component functions of ψ. Again, it is unkown whether the degree restriction can be omitted.

Let us outline the proof of Theorem 1.1. By the *straight-line complexity $L(g)$* of a multivariate polynomial g over k we understand the minimal number of arithmetic operations

---

sufficient to compute g by a straight-line program without divisions from the variables and constants in k. The *decision complexity* $C(g)$ of g is defined as the minimal number of arithmetic operations and tests sufficient for an algebraic computation tree to decide for given points x in k^n whether $g(x) = 0$. The decision complexity of the graph of φ considered above is formally defined as the decision complexity of the polynomial $Y - \varphi(X_1, \dots, X_n)$.

It is clear that $C(g) \leq L(g) + 1$. A sort of converse is provided by the following well-known lemma based on the Nullstellensatz (cf. [11, 3]).

Lemma 1.2 *Let g be the irreducible generator of a hypersurface in $\mathbb{R}^n$ or $\mathbb{C}^n$. Then there exists a nonzero multiple f of g such that $L(f) \leq C(g)$.*

Note that if the zeroset of g is not a hypersurface, then the conclusion becomes wrong (take $g = X_1^{2d} + \dots + X_n^{2d}$ over $\mathbb{R}$). Over the reals, we in fact need the assumption that g is irreducible (see [11] and the discussion below). We remark that the conclusion of this lemma remains true over any infinite field k if g is the generator of the graph of a polynomial φ, that is, $g = Y - \varphi(X_1, \dots, X_n)$.

This lemma leads naturally to the question of relating the complexity of a polynomial f to those of its factors. Unfortunately, there exist polynomials f having factors with a complexity exponential in the complexity of f, as first discovered by Lipton and Stockmeyer [24]. The simplest known example illustrating this is as follows: Consider $f_n = X^{2^n} - 1 = \prod_{j < 2^n} (X - \zeta^j)$, where $\zeta = \exp(2\pi i / 2^n)$. By repeated squaring we get $L(f_n) \leq n + 1$. On the other hand, one can prove that for almost all $M \subseteq \{0, 1, \dots, 2^n - 1\}$ the random factor $\prod_{j \in M} (X - \zeta^j)$ has a complexity which is exponential in n, cf. [9, Exercise 9.8]. A similar reasoning can be made over the rationals based on the factorization into the cyclotomic polynomials.

This idea yields reducible factors of high complexity. It is plausible that this effect may occur also for irreducible factors. This is an open problem, however, which is related to the question of whether the degree restriction in Theorem 1.1 can be omitted. Note that in the case $k = \mathbb{R}$ the following trivial example from [11] shows that $L(g)$ may be exponentially larger than $C(g)$. Let $g_n \in \mathbb{R}[X]$ have n distinct real roots. Then $C(g_n) \leq \log n$ using binary search, but $L(g_n) \geq n$ if the roots of g are algebraically independent over $\mathbb{Q}$.

In the above example by Lipton and Stockmeyer, the degree of the factor g is exponential in the complexity of f. We restrict now our attention to factors having a degree polynomially bounded in the complexity of f. Kaltofen [20] proved that the complexity of any irreducible factor g is polynomially bounded in the complexity of f and in the degree and the *multiplicity* of the factor g. Our *Factor Conjecture* [6, Conj. 8.3] claims that the dependence on the

multiplicity can be omitted. The main result of this paper (Theorem 4.1) states that the dependence on the multiplicity can indeed be omitted when switching to approximative complexity.

The idea of the proof of Theorem 4.1 is as follows: After a suitable coordinate transformation one can interpret the zeroset of the factor g locally around the origin as the graph of some analytic function φ. In order to cope with a possibly large multiplicity of g, we apply a small perturbation to the polynomial f without affecting its complexity too much. This results in a small perturbation of φ. We compute now the homogeneous parts of the perturbed φ by a Newton iteration up to a certain order. Using efficient polynomial arithmetic, this gives us an upper bound on the approximative complexity of the homogeneous parts of φ up to a predefined order (Proposition 4.3). In the special case, where the factor g is the generator of the graph of a function, we are already done. This is essentially the contents of Section 4.1.

In a second step, elaborated in Section 4.2, we view the factor g as the minimal polynomial of φ in the variable $Y := X_n$ over the field $k(X_1, \dots, X_{n-1})$. We show that the Taylor approximations up to order $2d^2$ uniquely determine the factor g and compute the bihomogeneous components of g with respect to the degrees in the X-variables and Y by fast linear algebra.

In Section 5 we discuss applications to the relationship between the complexity of decisional and computational tasks. There we also build in the concept of approximative complexity into Valiant's algebraic P-NP framework [31, 33] (see also [9, 6]) and make a connection to the Blum-Shub-Smale model of computation [4].

For some other aspects of the issues dicussed in this paper see [8].

2 Preliminaries

In what follows, $M(d)$ denotes an upper bound on the complexity for the multiplication of two univariate polynomials of degree d over k, i.e., for computing the coefficients of the product polynomial from the coefficients of the given polynomials. It is well known that $M(d) = O(d \log d)$ if the field k "supports fast Fourier transforms", for instance, if k is the field of complex numbers.

The following result is obtained by a technique introduced by Strassen [27] for the computation of homogeneous components and avoiding divisions.

Proposition 2.1 *Assume that $F(X,Y) = \sum_{i,\delta} F_i^{(\delta)} Y^i$ is the bihomogeneous decomposition of the polynomial $F \in k[X_1, \ldots, X_n, Y]$, thus $F_i^{(\delta)}$ is a homogeneous polynomial in the X-variables of degree δ. Then we have for all $D \geq 1$*

$$L(\{F_i^{(\delta)} \mid i, \delta \leq D\}) = O(M(D^2)L(F))$$

and the same is true if the complexity L is replaced by the approximative complexity $\underline{L}$ to be introduced in Section 3.

The next lemma follows immediately from the well-known algorithms for univariate power series described in [9, §2.4] by interpreting the homogeneous components of a multivariate power series $f \in k[[X_1, \ldots, X_n]]$ as the T-adic coefficients of the transformed series $f(TX_1, \ldots, TX_n)$.

Lemma 2.2 (1) *We can compute the homogeneous parts up to degree D of the product $F \cdot G$ and of the quotient F/G (if $G(0) \neq 0$) of multivariate power series F and G from the homogeneous parts of F and G up to degree D by $M(D)$ arithmetic operations.*

(2) *Assume that the multivariate power series $F_0, \ldots, F_D$ and Φ are given by their homogeneous parts up to degree D. Then we can compute from this data the homogeneous parts of $\sum_{i=0}^{D} F_i \Phi^i$ up to degree D by $O(DM(D))$ arithmetic operations.*

3 Approximative Complexity

In complexity theory it has proven useful to study "approximative algorithms", which use arithmetic with infinite precision and nevertheless only give us an approximation of the solution to be computed, however with any precision required. This concept was systematically studied in the framework of bilinear complexity (border rank) and there it has turned out to be one of the main keys to the currently best known fast matrix multiplication algorithms [12]. We refer to [9, Chap. 15] and the references there for further information.

Although approximative complexity is a very natural concept, it has been investigated in less detail for computations of polynomials or rational functions. Originally, it had been introduced by Strassen in a topological way [28]. Griesser [16] generalized most of the known lower bounds for multiplicative complexity to approximative complexity. Lickteig [22] and Grigoriev and Karpinski [17] employ the notion of approximative complexity for proving lower bounds.

It is not known how to meaningfully relate the complexity of leading coefficients or of factors of a polynomial to the complexity of the polynomial itself. However, by allowing approximative computations, we are able to establish quite satisfactory reductions in these cases. The deeper reason why this is possible seems to be the lower semicontinuity of the approximative complexity, which allows a controlled passage to the limit and can be used in perturbation arguments.

Assume the polynomial f is expanded with respect to Y:

$$f = f_q(X_1, \ldots, X_n)Y^q + f_{q+1}(X_1, \ldots, X_n)Y^{q+1} + \ldots.$$

We do not know whether the complexity of the leading coefficient f_q can be polynomially bounded in the the complexity of f. However, we can make the following observation. For the moment assume that k is the field of real or complex numbers. We have $\lim_{y \to 0} y^{-q} f(X, y) = f_q(X)$ and $L(f(X,y)) \leq L(f)$ for all $y \in k$. Thus we can approximate f_q with arbitrary precision by polynomials having complexity at most $L(f)$. We will say that f_q has "approximate complexity" at most $L(f)$.

In what follows, we will formalize this in an algebraic way; a topological interpretation will be given later. Throughout the paper, $K := k(\epsilon)$ is a rational function field in the indeterminate ϵ over the field k and R denotes the local subring of K consisting of the rational functions defined at $\epsilon = 0$. We write $F_{\epsilon=0}$ for the image of $F \in R[X]$ under the morphism $R[X] \to k[X]$ induced by $\epsilon \mapsto 0$.

Definition 3.1 Let $f \in k[X_1, \ldots, X_n]$. The *approximative complexity* $\underline{L}(f)$ of the polynomial f is the smallest natural number r such that there exists F in $R[X_1, \ldots, X_n]$ satisfying $F_{\epsilon=0} = f$ and $L(F) \leq r$. Here the complexity L is to be interpreted with respect to the larger field of constants K.

Even though L refers to division-free straight-line programs, divisions will occur implicitly since our model allows the free use of any elements of K as constants (e.g., division by powers of ϵ). In fact, the point is that even though F is defined with respect to the morphism $\epsilon \mapsto 0$, the intermediate results of the computation may not be so! Note that $\underline{L}(f) \leq L(f)$.

We remark that the assumption that any elements of K are free constants is just made for conceptual simplicity. We may as well require to build up the needed elements of K from ϵ, ϵ^{-1} and elements of k. It is easy to see that this would not change our main result (i.e., Theorem 4.1).

Assume that $\underline{L}(f) \leq r$ over $k = \mathbb{R}$, say $F_\epsilon(x) = f(x) + \epsilon R_\epsilon(x)$ and $L(F_\epsilon) \leq r$. Let M be the supremum of $R_\epsilon(x)$ over all $\epsilon \in [0,1]$ and $x \in \mathbb{R}^n$ with $||x||_\infty \leq 1$. Then we have for such ϵ and x that $|F_\epsilon(x) - f(x)| = \epsilon |R_\epsilon(x)| \leq M\epsilon$. Therefore, for each $\epsilon > 0$ we can compute on input x an approximation to $f(x)$ with absolute error less than $M\epsilon$ with only r arithmetic operations. If we would additionally require in the definition of $\underline{L}$ to build up the needed constants in K from ϵ, ϵ^{-1}, then $\underline{L}(f) \leq r$ would even mean that one can compute an approximation with error less than $M\epsilon$ with only r arithmetic operations on input x *and* ϵ.

We omit the proof of the basic properties of $\underline{L}$ listed in the remark below. (For part (3) compare [6, Prop. 4.1(iii)].)

Remark 3.2 1. (*Semicontinuity*) The quantity $\underline{L}(F)$ is clearly defined for a polynomial F over K (by adjoining a further indeterminate to K). If F is defined over R and $f = F_{\epsilon=0}$, then $\underline{L}(f) \leq \underline{L}(F)$.

2. (*Elimination of constants*) Let k_1 be a field extension of k of degree at most d and f be a polynomial over k. Then $\underline{L}(f) = O(M(d)\,\underline{L}_{k_1}(f))$, where $\underline{L}_{k_1}(f)$ denotes the approximative complexity of f interpreted as a polynomial over k_1 (i.e., constants in k_1 may be used freely).

3. (*Transitivity*) The approximative complexity $\underline{L}(f \mid g)$ to compute f from g and the variables is defined in a natural way. We have $\underline{L}(f) \leq \underline{L}(f \mid g) + \underline{L}(g)$, and a corresponding observation is true for the computation of several polynomials.

We proceed with a topological interpretation of approximative complexity, which points out the naturality of this notion from a mathematical point of view. However, this comment will not be needed in the remainder of the paper. Assume k to be an algebraically closed field. There is a natural way to put a Zariski topology on the polynomial ring $A_n := k[X_1, \ldots, X_n]$ as a limit of the Zariski topologies on the finite dimensional subspaces $\{f \in A_n \mid \deg f \leq d\}$ for $d \in \mathbb{N}$. If k is the field of complex numbers, we may define the Euclidean topology on A_n in a similar way.

If $f \in A_n$ satisfies $\underline{L}(f) \leq r$, then it easy to see that f lies in the closure (Zariski or Euclidean) of the set $\{f \in A_n \mid L(f) \leq r\}$. Indeed, we have $L(F_{\epsilon=y}) \leq L(F)$ for all but finitely many $y \in k$ and $\lim_{y \to 0} F_{\epsilon=y} = F_{\epsilon=0} = f$. Alder [1] has shown that the converse is true and obtained the following topological characterization of the approximative complexity.

Theorem 3.3 *The set $\{f \in A_n \mid \underline{L}(f) \leq r\}$ is the closure of the set $\{f \in A_n \mid L(f) \leq r\}$ for the Zariski topology. If $k = \mathbb{C}$, this is also true for the Euclidean topology.*

4 Approximative Complexity of Factors

Let γ be strictly larger than the exponent ω of matrix multiplication, thus we assume that n by n matrices can be multiplied with $O(n^\gamma)$ arithmetic operations. (It is known that $2 \leq \omega < 2.38$, see [9, Chap. 15].)

Here is the main result of this paper.

Theorem 4.1 *Let k be a field of characteristic zero and assume that g is an irreducible factor of degree d and multiplicity e of a polynomial $f \in k[X_1, \ldots, X_n]$. Then*

$$\underline{L}(g) = O\big(M(d)M(d^4)\underline{L}(f) + d^{2\gamma}M(d)^2\big).$$

Remark 4.2 There is room for improvement in this bound. In fact, the proof of Theorem 4.1 yields better estimates in the following cases. If g is the generator of the graph of a polynomial function, we obtain $\underline{L}(g) = O\big(M(d^2)\underline{L}(f)\big)$. In the case $k = \mathbb{R}$ or $\mathbb{C}$, we get $\underline{L}(g) = O\big(M(d^4)\underline{L}(f) + d^{2\gamma}M(d)\big)$.

The proof of Theorem 4.1 will be supplied in the next two sections.

4.1 Approximative Computation of Graph

We assume that we are in the situation of Theorem 4.1 and write $f = g^e h$. Thus g is irreducible and g and h are coprime. It is convenient to use the notations $Y := X_n$ and $X := (X_1, \ldots, X_{n-1})$. We are first going to transform the polynomials into a special form by suitable linear transformations.

In some field extension k_1 of degree at most d over k there exists a root $(\xi, \eta) \in k_1^n$ of g, such that h as well as the gradient of g do not vanish there. To simplify notation, we assume that $k_1 = k$. This assumption will be eliminated at the end of Section 4.2 at the price of an additional factor $M(d)$ in the complexity bound.

By a coordinate shift we can always achieve that $(\xi, \eta) := (0, 0)$. By a substitution $\tilde{g}(X, Y) := g(X_1 + u_1 Y, \ldots, X_{n-1} + u_{n-1}Y, vY)$ we may achieve that the degree of $\tilde{g}$ in Y equals d and that $\partial_Y \tilde{g}(0, 0)$ does not vanish. Indeed, if $g^{(d)}$ denotes the homogeneous component of g of degree d, then the coefficient of Y^d in $\tilde{g}$ equals $\tilde{g}^{(d)}(0, 1) = g^{(d)}(u, v)$. Moreover, $\partial_Y \tilde{g}(0, 0) = u_1 \partial_{X_1} g(0, 0) + \ldots + u_{n-1} \partial_{X_{n-1}} g(0, 0) + v \partial_Y g(0, 0)$. Hence it suffices to choose u, v such that this linear combination does not vanish and such that $g^{(d)}(u, v) \neq 0$. By scaling, we may assume without loss of generality that $\tilde{g}$ is monic with respect to Y. In the following, we will assume that this transformation has already been done, i.e., $\tilde{g} = g$, which results in a complexity increase of f of at most $2n$. Note that $L(f) \geq n$ if all the variables occur in f.

Summarizing, we achieved the following by a suitable choice of a linear transformation:

$$g(0,0) = 0, \ h(0,0)\, \partial_Y g(0,0) \neq 0, \ \deg_Y g = d. \quad (1)$$

The implicit function theorem implies that there exists a unique formal power series $\varphi \in k[[X]]$ such that

$$g(X, \varphi(X)) = 0, \ \varphi(0) = 0. \quad (2)$$

Moreover, this power series can be recursively computed by the following *Newton iteration*: if we put $\varphi_0 = 0$ and define

$$\varphi_{\nu+1} = \varphi_\nu - \frac{g(X, \varphi_\nu)}{\partial_Y g(X, \varphi_\nu)}, \quad (3)$$

then we have quadratic convergence of the φ_ν towards φ, in the sense that $\varphi_\nu \equiv \varphi \bmod (X)^{2^\nu}$, where (X) denotes the maximal ideal of $k[[X]]$ (cf. [9, Theorem 2.31]).

It is easy to see that if the partial derivative $\partial_Y f(0,0)$ would not vanish, then the above power series φ could also be recursively computed by the Newton recursion (3) with g replaced by f. However, $\partial_Y f(0,0) = 0$ always vanishes for multiplicities $e > 1$. The key idea is now to enforce the nonvanishing of this partial derivative by a suitable perturbation of the given polynomial f. By doing so, we have to content ourselves with an approximative computation of the factor g.

Based on these ideas, we prove the following assuming the conditions (1):

Proposition 4.3 *The homogeneous parts $\varphi^{(\delta)}$ of φ of degree δ satisfy*

$$\forall D \geq 1: \quad \underline{L}(\varphi^{(1)}, \ldots, \varphi^{(D)}) = O\big(M(D^2)\underline{L}(f)\big).$$

Proof. Note that g, viewed as a polynomial in Y over $k(X)$, is the minimal polynomial of φ over $k(X)$. W.l.o.g. we may assume that φ is not a rational function (otherwise $d = 1$ and φ would be linear).

We define the perturbed polynomial $F(X, Y) := f(X, Y + \epsilon) - f(0, \epsilon)$ over the coefficient ring R. It is clear that $F(0,0) = 0$ and $F_{\epsilon=0} = f$. By a straight-forward calculation we get

$$\partial_Y F(0,0) = \big(eh\,\partial_Y g + g\,\partial_Y h\big)(0, \epsilon) \cdot g^{e-1}(0, \epsilon).$$

Assumptions (1) tell us that $g(0, \epsilon) = \lambda \epsilon + O(\epsilon^2)$ with $\lambda \in k^\times$, hence

$$\partial_Y F(0,0) = e\,\lambda^e\,h(0,0)\,\epsilon^{e-1} + O(\epsilon^e)$$

and we conclude that this partial derivative does not vanish ($\mathrm{char}\,k = 0$).

As in the reasoning before, the implicit function theorem implies that there exists a unique formal power series Φ over the field $K = k(\epsilon)$ such that $F(X, \Phi(X)) = 0$, $\Phi(0) = 0$ and this power series can be recursively computed by the Newton iteration

$$\Phi_0 = 0, \quad \Phi_{\nu+1} = \Phi_\nu - \frac{F(X, \Phi_\nu)}{\partial_Y F(X, \Phi_\nu)} \qquad (4)$$

with quadratic convergence: $\Phi_\nu \equiv \Phi \bmod (X)^{2^\nu}$.

Claim: Φ_ν is defined over the coefficient ring R for all ν.

We prove this claim by induction on ν, the induction start $\nu = 0$ being clear. So let us assume that Φ_ν is defined over R and set $\psi_\nu := (\Phi_\nu)_{\epsilon=0}$. By applying the morphism $R[[X]] \to k[[X]], \epsilon \mapsto 0$ we obtain

$$(\partial_Y F(X, \Phi_\nu))_{\epsilon=0} = \partial_Y f(X, \psi_\nu)$$
$$= \big(eh\,\partial_Y g + g\,\partial_Y h\big)(X, \psi_\nu) \cdot g^{e-1}(X, \psi_\nu).$$

The first parenthesis maps under the substitution $X \mapsto 0$ to $(eh\,\partial_Y g)(0,0)$, which is nonzero by our assumptions. The second factor $g(X, \psi_\nu)$ can only vanish if $\psi_\nu = \varphi$ since the power series φ is uniquely determined by the conditions (2). In this case, φ would be a rational function, which we have excluded at the beginning of the proof. We have thus shown that $(\partial_Y F(X, \Phi_\nu))_{\epsilon=0}$ nonzero. By equation (4) this implies that $\Phi_{\nu+1}$ is defined over R and proves the claim.

The claim implies that Φ is defined over R. From $F(X, \Phi(X)) = 0$ we obtain $f(X, (\Phi(X))_{\epsilon=0}) = 0$, hence $g(X, (\Phi(X))_{\epsilon=0}) = 0$, as $h(X, (\Phi(X))_{\epsilon=0}) \neq 0$. We conclude that $(\Phi(X))_{\epsilon=0} = \varphi$. If $\Phi^{(\delta)}$ denotes the homogeneous part of Φ of degree δ, we have $(\Phi_\nu)^{(\delta)} = \Phi^{(\delta)}$ for $\delta < 2^\nu$. This implies for $\delta < 2^\nu$ that

$$((\Phi_\nu)^{(\delta)})_{\epsilon=0} = (\Phi^{(\delta)})_{\epsilon=0} = (\Phi_{\epsilon=0})^{(\delta)} = \varphi^{(\delta)}.$$

As a word of warning, we point out that a certain care in these argumentations is necessary. For instance, examples show that in general $(\Phi_\nu)_{\epsilon=0} \neq \varphi_\nu$.

We turn now to the algorithmic analysis of the proof. First of all we note that $L(F) \leq L(f) + 2$. A moment's thought shows that also $\underline{L}(F) \leq \underline{L}(f) + 2$. In order to prove the proposition it is enough to show that

$$L(\Phi_N^{(1)}, \ldots, \Phi_N^{(D)}) = O\big(M(D^2)\underline{L}(f)\big), \qquad (5)$$

where $N := \lceil \log(D + 1) \rceil$. In fact, by the semicontinuity of $\underline{L}$, we only need to prove this estimate for approximative complexity on the lefthand side.

The following computation deals with polynomials in the X-variables, which are truncated at a certain degree and represented by their homogeneous parts up to this degree. We obtain from Proposition 2.1 for the bihomogeneous decomposition of F that

$$\underline{L}(\{F_i^{(\delta)} \mid i, \delta \leq D\}) = O\big(M(D^2)\underline{L}(F)\big). \qquad (6)$$

In the following, we assume that we have already computed the bihomogeneous components $F_i^{(\delta)}$ for $i, \delta \leq D$.

Inductively, we suppose that we have computed the homogeneous parts of Φ_ν up to degree 2^ν. The main work of one Newton step (4) consists in the computation of the substituted polynomials $F(X, \Phi_\nu)$ and $\partial_Y F(X, \Phi_\nu)$. By Lemma 2.2 we can compute the homogeneous parts up to degree $2^{\nu+1}$ of $F(X, \Phi_\nu)$ by $O(2^\nu M(2^\nu))$ arithmetic operations. Analogously, we get the homogeneous parts up to degree $2^{\nu+1}$ of $\partial_Y F(X, \Phi_\nu)$ by the same number of arithmetic operations. By a division and a subtraction we obtain from this the homogeneous parts of $\Phi_{\nu+1}$ up to degree $2^{\nu+1}$ using further $O(M(2^\nu))$ arithmetic operations. Altogether, we obtain

$$L(\Phi_N^{(1)}, \ldots, \Phi_N^{(D)} \mid \{F_i^{(\delta)} \mid i, \delta \leq D\})$$
$$= O\big(\sum_{\nu=0}^{N} 2^\nu M(2^\nu)\big) = O(DM(D)) = O(M(D^2)),$$

since $\sum_{\nu=0}^{N} M(2^\nu) \leq 2M(2^N)$ and $DM(D) \leq M(D^2)$ (see [9, Rem. 2.10]). The assertion (5) follows from this estimate and equation (6) by the transitivity of approximative complexity. $\qquad\square$

4.2 Reconstruction of Minimal Polynomial

Consider the bihomogeneous decomposition $g(X,Y) = \sum_{i,\alpha \leq d} g_i^{(\alpha)} Y^i$. Let T be an additional indeterminate and perform the substitution $X_j \mapsto T X_j$. The condition $g(X, \varphi(X)) = 0 \mod (X)^{D+1}$ translates to

$$\sum_{i,\alpha \leq d} g_i^{(\alpha)} T^\alpha \big(\sum_{\delta \leq D} \varphi^{(\delta)} T^\delta \big)^i \equiv 0 \mod T^{D+1}$$

for any $D \geq 1$. Moreover, we have $g_d^{(0)} = 1$ and $g_d^{(\alpha)} = 0$ for $0 < \alpha \leq d$, since g is monic of degree d in Y. The next lemma states that these conditions uniquely determine the bihomogeneous components of g if we choose $D \geq d^2$. The proof is based on well-known ideas from the analysis of the LLL-algorithm [21] (see also [15, Lemma 16.20]) adapted from $\mathbb{Z}$ to the setting of a polynomial ring.

Lemma 4.4 *By comparing the coefficients of the powers of the indeterminate T, one can interpret the conditions*

$$\sum_{i,\alpha \leq d} Z_{i,\alpha} T^\alpha \big(\sum_{\delta \leq 2d^2} \varphi^{(\delta)} T^\delta \big)^i \equiv 0 \mod T^{2d^2+1},$$
$$Z_{d,0} = 1, Z_{d,1} = 0, \ldots, Z_{d,d} = 0$$

as a system of linear equations over the field $k(X)$ in the unknowns $Z_{i,\alpha}$. (There are $2d^2 + 1$ equations and $(d+1)^2$ unknowns). This linear system has as the unique solution the bihomogeneous components $Z_{i,\alpha} = g_i^{(\alpha)}$ of g.

Proof. We define the bivariate polynomial $A(T,Y) := \sum_{i,\alpha \leq d} g_i^{(\alpha)} T^\alpha Y^i$ over $k(X)$ and assign to a solution $\zeta_{i,\alpha}$ of the above linear system of equations the bivariate polynomial $B(T,Y) := \sum_{i,\alpha \leq d} \zeta_{i,\alpha} T^\alpha Y^i$. Note that A is an irreducible polynomial in Y over $k(X,T)$ since we assume g to be irreducible and monic with respect to Y. The polynomial $\psi := \sum_{\delta \leq 2d^2} \varphi^{(\delta)} T^\delta$ is an approximative common root of A and B in the sense that

$$A(T,\psi) \equiv 0 \mod T^{2d^2+1}, \quad B(T,\psi) \equiv 0 \mod T^{2d^2+1}.$$

The resultant $\mathrm{res}(A,B) \in k(X)[T]$ of A and B with respect to Y satisfies the degree estimate

$$\deg_T \mathrm{res}(A,B)$$
$$\leq \deg_Y A \cdot \deg_T B + \deg_T A \cdot \deg_Y B \leq 2d^2,$$

which is easily seen from the description of the resultant as the determinant of the Sylvester matrix (cf. [15, §6.3]). It is well-known that there exist polynomials $u, v \in k(X)[T, Y]$

such that $uA + vB = \mathrm{res}(A,B)$. Substituting the approximative common root ψ for Y in this equation implies that $\mathrm{res}(A,B) \equiv 0 \mod T^{2d^2+1}$, hence the resultant vanishes. Since A is irreducible, it must be a factor of B over $k(X,T)$. However, we assume A and B both to be monic with respect to Y. This implies that in fact $A = B$ as claimed. $\qquad\square$

The coefficients of the linear system of equations in Lemma 4.4 can be computed from the homogenous components $\varphi^{(\delta)}$, $\delta \leq 2d^2$, with d multiplications of power series given by their coefficients up to degree $2d^2$. This can be done with $O(dM(d^2))$ arithmetic operations (Lemma 2.2). If γ denotes a number strictly larger than the exponent ω of matrix multiplication, then we can compute from the coefficients of the linear system the unique solution with $O(d^{2\gamma})$ operations (see [9, Chap. 16]). This computation can be interpreted as a straight-line program involving divisions. However, as the bihomogeneous components of g we are seeking for are homogenous of degree at most d, we can apply Strassen's idea of avoiding divisions [27] and transform this straight-line program into one without divisions, which is at most by a factor of $O(M(d))$ longer. Summarizing, we obtain the following:

Proposition 4.5 *We have for any $\gamma > \omega$ that*

$$L\big(\{ g_i^{(\delta)} \mid i, \delta \leq d \} \mid \varphi^{(1)}, \ldots, \varphi^{(2d^2)} \big) = O\big(d^{2\gamma} M(d) \big).$$

Our main Theorem 4.1 is a consequence of this Proposition and Proposition 4.3. In fact, this provides an upper bound on $\underline{L}(g)$ with respect to the field extension k_1 of k of degree at most d considered at the beginning of Section 4.1. To simplify notation, we assumed there that $k_1 = k$. This assumption can now be eliminated at the price of an additional factor $M(d)$ in the complexity bound according to Remark 3.2(2).

5 Applications to Decision Complexity

By combining Theorem 4.1, Remark 4.2, and Lemma 1.2, we obtain the following corollary.

Corollary 5.1 *Let g be the generator of an irreducible hypersurface in $\mathbb{R}^n$ or $\mathbb{C}^n$, $d = \deg g$, and γ be strictly larger than the exponent ω of matrix multiplication. Then we have*

$$\underline{L}(g) = O\big(M(d^4) C(g) + d^{2\gamma} M(d) \big).$$

This implies Theorem 1.1 of the introduction for decision complexity. The claim about randomized complexity (formalized by randomized algebraic computation trees) follows easily from Corollary 5.1 by the results in [25, 10, 13]. We remark that if the hypersurface is the graph of a polynomial function, then we obtain the better bound $\underline{L}(g) = O\big(M(d^2) C(g) \big)$.

In [31, 33] Valiant had proposed an analogue of the theory of NP-completeness in a framework of algebraic complexity, in connection with his famous hardness result for the permanent [32]. This theory features algebraic complexity classes VP and VNP as well as VNP-completeness results for many families of generating functions of graph properties, the most prominent being the family of permanents. There is rather strong evidence for Valiant's hypothesis VP $\neq$ VNP. In fact, if it were false, then the nonuniform versions of the complexity classes NC and PH would collapse [7]. For a comprehensive presentation of this theory, we refer to [14, 9, 6]. In the following, we assume some basic familiarity with the concepts introduced there.

It is quite natural to incorporate the concept of approximative complexity into Valiant's framework.

Definition 5.2 An *approximatively p-computable family* is a *p*-family (f_n) such that $\underline{L}(f_n)$ is a *p*-bounded function of *n*. The complexity class $\underline{VP}$ comprises all such families over a fixed field *k*.

It is obvious that VP $\subseteq$ $\underline{VP}$. If the polynomial *f* is a projection of a polynomial *g*, then we clearly have $\underline{L}(f) \leq \underline{L}(g)$. Therefore, the complexity class $\underline{VP}$ is closed under *p*-projections. We remark that $\underline{VP}$ is also closed under the polynomial oracle reductions introduced in [5].

We know very few about the relations between the complexity classes VP, $\underline{VP}$, and VNP. Intuitively, one would think that $\underline{VP}$ should not differ too much from VP. Likewise, it could be that the class $\underline{VP}$ is contained in VNP. The hypothesis

$$\text{VNP} \not\subseteq \underline{\text{VP}} \tag{7}$$

is a strengthening of Valiant's hypothesis, which is equivalent to saying that VNP-complete families are not approximately *p*-computable.

This hypothesis should be compared with the known work on polynomial time deterministic or randomized approximation algorithms for the permanent of non-negative matrices [23, 2, 19]. Based on the Markov chain approach, Jerrum, Sinclair and Vigoda [19] have recently established a fully-polynomial randomized approximation scheme for computing the permanent of an arbitrary real matrix with non-negative entries. We note that this result does not contradict hypothesis (7), since the above mentioned algorithm works only for matrices with *non-negative* entries, while approximative straight-line programs a fortiori work on all real inputs.

Under the hypothesis VNP $\not\subseteq$ $\underline{VP}$, we can can conclude that checking the values of polynomials forming VNP-complete families is hard, even when we allow randomized algorithms with two-sided error.

Corollary 5.3 *Assume* VNP $\not\subseteq$ $\underline{VP}$ *over a field k of characteristic zero. Then for any* VNP-*complete family* (g_n),

checking the value $y = g_n(x)$ cannot be done by deterministic or randomized algebraic computation trees with a polynomial number of arithmetic operations and tests in n.

Hypothesis (7) implies a separation of complexity classes in the Blum-Shub-Smale model of computation [4]. See [3] for the definition of the classes $P_{\mathbb{R}}$ and $PAR_{\mathbb{R}}$. (For the proof use Corollary 5.1 with the permanent polynomial g.)

Corollary 5.4 *If* VNP $\not\subseteq$ $\underline{VP}$ *is true, then we have* $P_{\mathbb{R}} \neq PAR_{\mathbb{R}}$ *in the Blum-Shub-Smale model over the reals.*

References

[1] A. Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht.* PhD thesis, Zürich University, 1984.

[2] A. Barvinok. Polynomial time algorithms to approximate permanents and mixed discriminants within a simply exponential factor. *Random Structures and Algorithms*, 14:29–61, 1999.

[3] L. Blum, F. Cucker, M. Shub, and S. Smale. *Complexity and Real Computation.* Springer, 1998.

[4] L. Blum, M. Shub, and S. Smale. On a theory of computation and complexity over the real numbers. *Bull. Amer. Math. Soc.*, 21:1–46, 1989.

[5] P. Bürgisser. On the structure of Valiant's complexity classes. *Discr. Math. Theoret. Comp. Sci.*, 3:73–94, 1999.

[6] P. Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and Computation in Mathematics*. Springer Verlag, 2000.

[7] P. Bürgisser. Cook's versus Valiant's hypothesis. *Theoret. Comp. Sci.*, 235:71–88, 2000.

[8] P. Bürgisser. On implications between P-NP-hypotheses: Decision versus computation in algebraic complexity. In *Proc. 26th MFCS*, LNCS. Springer Verlag, 2001. To appear.

[9] P. Bürgisser, M. Clausen, and M. Shokrollahi. *Algebraic Complexity Theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer Verlag, 1997.

[10] P. Bürgisser, M. Karpinski, and T. Lickteig. On randomized semialgebraic decision complexity. *J. Compl.*, 9:231–251, 1993.

[11] P. Bürgisser, T. Lickteig, and M. Shub. Test complexity of generic polynomials. *J. Compl.*, 8:203–215, 1992.

[12] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *J. Symb. Comp.*, 9:251–280, 1990.

[13] F. Cucker, M. Karpinski, P. Koiran, T. Lickteig, and K. Werther. On real Turing machines that toss coins. In *Proc. 27th ACM STOC, Las Vegas*, pages 335–342, 1995.

[14] J. v. z. Gathen. Feasible arithmetic computations: Valiant's hypothesis. *J. Symb. Comp.*, 4:137–172, 1987.

[15] J. v. z. Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[16] B. Griesser. Lower bounds for the approximative complexity. *Theoret. Comp. Sci.*, 46:329–338, 1986.

[17] D. Grigoriev and M. Karpinski. Randomized quadratic lower bound for knapsack. In *Proc. 29th ACM STOC*, pages 76–85, 1997.

[18] J. Grollmann and A. Selman. Complexity measures for public-key cryptosystems. *SIAM J. Comp.*, 17(2):309–335, 1988.

[19] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with non-negative entries. *Electronic Colloquium on Computational Complexity*, 2000. Report No. 79.

[20] E. Kaltofen. Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proc. 19th ACM STOC*, pages 443–452, 1986.

[21] A. Lenstra, H. L. Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Math. Ann.*, 261:515–534, 1982.

[22] T. Lickteig. On semialgebraic decision complexity. Technical Report TR-90-052, Int. Comp. Sc. Inst., Berkeley, 1990. Habilitationsschrift, Universität Tübingen.

[23] N. Linial, A. Samorodnitsky, and A. Wigderson. A deterministic polynomial algorithm for matrix scaling and approximate permanents. In *Proc. 30th ACM STOC*, pages 644–652, 1998.

[24] R. Lipton and L. Stockmeyer. Evaluation of polynomials with super-preconditioning. *J. Comp. Syst. Sci.*, 16:124–139, 1978.

[25] F. Meyer auf der Heide. Simulating probabilistic by deterministic algebraic computation trees. *Theoret. Comp. Sci.*, 41:325–330, 1985.

[26] A. Selman. A survey of one-way functions in complexity theory. *Math. Systems Theory*, 25:203–221, 1992.

[27] V. Strassen. Vermeidung von Divisionen. *Crelles J. Reine Angew. Math.*, 264:184–202, 1973.

[28] V. Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM J. Comp.*, 3:128–149, 1974.

[29] C. Sturtivant and Z. L. Zhang. Efficiently inverting bijections given by straight line programs. In *Proc. 31th FOCS*, pages 327–334, 1990.

[30] L. Valiant. Relative complexity of checking and evaluating. *Inf. Proc. Letters*, 5:20–23, 1976.

[31] L. Valiant. Completeness classes in algebra. In *Proc. 11th ACM STOC*, pages 249–261, 1979.

[32] L. Valiant. The complexity of computing the permanent. *Theoret. Comp. Sci.*, 8:189–201, 1979.

[33] L. Valiant. Reducibility by algebraic projections. In *Logic and Algorithmic: an International Symposium held in honor of Ernst Specker*, volume 30, pages 365–380. Monogr. No. 30 de l'Enseign. Math., 1982.