SCHOOL OF OPERATIONS RESEARCH
AND INDUSTRIAL ENGINEERING
COLLEGE OF ENGINEERING
CORNELL UNIVERSITY
ITHACA, NY 14853-3801

TECHNICAL REPORT NO. 957

February 1991
(Revised May 1991)

**Recent Progress on the
Complexity of the
Decision Problem
for the Reals¹**

by

James Renegar

¹Research was supported by the National Science Foundation grant DMS-8800835.

Recent Progress on the Complexity of the Decision Problem for the Reals*

James Renegar
School of Operations Research
and Industrial Engineering
Engineering and Theory Center
Cornell University
Ithaca, NY 14853

May 1991

Abstract

This paper concerns recent progress on the computational complexity of decision methods and quantifier elimination methods for the first order theory of the reals. The paper begins with a quick introduction of terminology followed by a short survey of some complexity highlights. We then discuss ideas leading to the most (theoretically) efficient algorithms known. The discussion is necessarily simplistic as a rigorous development of the algorithms forces one to consider a myriad of details.

This paper is similar to a talk the author gave at the DIMACS workshop on Algebraic Methods in Geometric Computations, held in May, 1990. A complete development can be found in [17].

*This research was supported by NSF Grant No. DMS-8800835.
1991 Mathematics Subject Classification. Primary 03, 12, 14, 68.

1 Some Terminology

A *sentence* is an expression composed from certain ingredients. Letting $\mathbb{R}$ denote a real-closed field, the following is an example of a sentence:

$$(\exists x_1 \in \mathbb{R}^{n_1})(\forall x_2 \in \mathbb{R}^{n_2})[(g_1(x_1, x_2) > 0) \vee (g_2(x_1, x_2) = 0)] \wedge (g_3(x_1, x_2) \neq 0). \quad (1.1)$$

The ingredients are: vectors of variables (x_1 and x_2); the quantifiers $\exists$ and $\forall$; atomic predicates (e.g. $g_1(x_1, x_2) > 0$) which are real polynomial inequalities ($>, \geq, =, \neq, \leq, <$); and a Boolean function holding the atomic predicates ($[B_1 \vee B_2] \wedge B_3$).

A sentence asserts something. The above sentence asserts that there exists $x_1 \in \mathbb{R}^{n_1}$ such that for all $x_2 \in \mathbb{R}^{n_2}$, (i) either $g_1(x_1, x_2) > 0$ or $g_2(x_1, x_2) = 0$, and (ii) $g_3(x_1, x_2) \neq 0$. Depending on the specific coefficients of the atomic predicate polynomials this assertion is either true or false.

The set of all true sentences constitutes the first order theory of the reals. A *decision method* for the first order theory of the reals is an algorithm that, given any sentence, correctly determines if the sentence is true. Decision methods for the reals were first proven to exist by Tarski [18], who constructed one.

A sentence is a special case of a more general expression, called a *formula*. Here is an example of a formula:

$$(\exists x_1 \in \mathbb{R}^{n_1})(\forall x_2 \in \mathbb{R}^{n_2})[(g_1(z, x_1, x_2) > 0) \vee (g_2(z, x_1, x_2) = 0)] \wedge (g_3(z, x_1, x_2) \neq 0). \quad (1.2)$$

A formula has one thing that a sentence does not, namely, a vector $z \in \mathbb{R}^{n_0}$ of *free variables*. When specific values are substituted for the free variables, the formula becomes a sentence.

A vector $\bar{z} \in \mathbb{R}^{n_0}$ is a *solution* for the formula if the sentence obtained by substituting $\bar{z}$ is true.

Two formulae are *equivalent* if they have the same solutions.

A *quantifier elimination method* is an algorithm that, given any formula, computes an equivalent quantifier-free formula, i.e., for the above formula $(\exists x_1 \in \mathbb{R}^{n_1})(\forall x_2 \in \mathbb{R}^{n_2})P(z, x_1, x_2)$ such a method would compute an equivalent formula $Q(z)$ containing no quantified variables.

When a quantifier elimination method is applied to a sentence, it becomes a decision method. Thus, a quantifier elimination method is in some sense more general than a decision method.

Tarski [18] actually constructed a quantifier elimination method.

Both (1.1) and (1.2) are said to be in *prenex* form, i.e., all quantifiers occur in front. More generally, a formula can be constructed from other formulae just as (1.1) was constructed from the atomic predicates, i.e., we can construct Boolean combinations of formulae, then quantify any or all of the free variables. If we quantify all free variables then we obtain a sentence.

2 Some Complexity Highlights

We now present a brief survey of some complexity highlights for quantifier elimination methods, considering only formulae in prenex form. General bounds follow inductively. (If a formula is constructed from other formulae, first apply quantifier elimination to the innermost formulae, then to the innermost formulae of the resulting formula, etc.)

We consider the general formula

$$(Q_1 x_1 \in \mathbb{R}^{n_1}) \cdots (Q_\omega x_\omega \in \mathbb{R}^{n_\omega}) P(z, x_1, \dots, x_\omega) \quad (2.1)$$

where $Q_1, \dots, Q_\omega$ are quantifiers, assumed without loss of generality to alternate, i.e., Q_i is not the same as Q_{i+1} . Let m denote the number of distinct polynomials occurring among the atomic predicates and let $d \geq 2$ be an upper bound on their degrees.

In the case of Turing machine computations where all polynomial coefficients are restricted to be integers, we let L denote the maximal bit length of the coefficients. In this context we refer to the number of *bit operations* required by a quantifier elimination method. In the general and idealized case that the coefficients are real numbers we rely on the computational model of Blum, Shub and Smale [2], and refer to *arithmetic operations*, these essentially being field operations, including comparisons.

The sequential bit operation bounds that have appeared in the literature are all basically of the form

$$(md)^E [L^{O(1)} + Cost] \quad (2.2)$$

where E is some exponent and $Cost$ is the worst-case cost of evaluating the Boolean function holding the atomic predicates, i.e., worst-case over 0-1 vectors. The $Cost$ term is generally relatively negligible compared to the factor $(md)^E$. Recent complexity bound improvements concern the exponent E .

The first reasonable upper bound for a quantifier elimination method was proven around 1973 by Collins [7]. He obtained $E = 2^{O(n)}$ where $n := n_0 + \dots + n_\omega$. Collins' bound is thus *doubly exponential* in the number of variables. His method requires the formula coefficients to be integers, the number of arithmetic operations (not just bit operations) growing with the size of the integers. Collins' algorithm was not shown to parallelize, although enough is now known from work of Neff [15] that a parallel version probably could be developed. Collins' work has been enormously influential in the area.

The next major complexity breakthrough was made around 1985 by Grigor'ev [12] who developed a decision method for which $E \approx [O(n)]^{4\omega}$. Grigor'ev's bound is doubly exponential only in the number of quantifier alternations. Many interesting problems can be cast as sentences with only a few quantifier alternations. For these, Grigor'ev's result is obviously significant. Like Collins' quantifier elimination method, Grigor'ev's decision method requires integer coefficients and was not proven to completely parallelize.

Slightly incomplete ideas of Ben-Or, Kozen and Reif [1] (approximately 1986) were completed by Fitchas, Galligo and Morgenstern [11] (approximately 1987) to construct a quantifier elimination method with arithmetic operation bound

$$(md)^E Cost \tag{2.3}$$

where $E = 2^{O(n)}$. This provides an arithmetic operation analog of Collins' bit operation bound. When restricted to integer coefficients, the method also yields the Collins' bound if the arithmetic operations are carried out bit by bit. Moreover, the algorithm parallelizes. Assuming each arithmetic operation requires one time unit, the resulting time bound is

$$[E \log(md)]^{O(1)} + Time(N) \tag{2.4}$$

if $(md)^E N$ parallel processors are used, where $Time(N)$ is the worst-case time required to evaluate the Boolean function holding the atomic predicates

using N parallel processors. The analogous time bound for bit operations is also valid, namely, $[E \log(Lmd)]^{O(1)} + \text{Time}(N)$ if $(md)^E [L^{O(1)} + N]$ parallel processors are used.

In [17] (1989), the author introduced a new quantifier elimination method for which $E = \prod_{k=0}^{\omega} O(n_k)$. This was established for arithmetic operations and bit operations, i.e., (2.3) and (2.2). Regarding bit operations, the dependence of the bounds on L was shown to be very low, $L^{O(1)}$ in (2.2) being replaced by $L(\log L)(\log \log L)$, i.e., the best bound known for multiplying two L -bit integers. Moreover, the method was shown to parallelize, resulting in the arithmetic operation time bound (2.4) if $(md)^E N$ parallel processors are used, and the bit operation time bound $\log(L)[E \log(md)]^{O(1)} + \text{Time}(N)$ if $(md)^E [L^2 + N]$ parallel processors are used.

Independently and simultaneously, Heintz, Roy and Solernó [14] developed a quantifier elimination method for which $E = O(n)^{O(\omega)}$, both for arithmetic and bit operations. Their method also completely parallelizes.

The various bounds are best understood by realizing that quantifier elimination methods typically work by passing through a formula from back to front. First the vector x_{ω} is focused on, then the vector $x_{\omega-1}$, and so on. The work arising from each vector results in a factor for E . For Collins' quantifier elimination method, the factor corresponding to x_k is $2^{O(n_k)}$ (note that $2^{O(n)} = 2^{O(n_0)} \dots 2^{O(n_{\omega})}$). For the method introduced by the author, the factor is $O(n_k)$. The factor corresponding to Grigor'ev's decision method is approximately $O(n)^4$ independently of the number of variables in x_k . In that method a vector with few variables can potentially create as large of a factor as one with many variables. Similarly, the factor corresponding to the quantifier elimination method of Heintz, Roy and Solernó is $O(n)^{O(1)}$ independently of x_k .

Many interesting formulae have blocks of variables of various sizes. For example, sentences asserting continuity generally have $(\forall \epsilon)(\exists \delta)$, along with large blocks of variables. The exponent $E = \prod O(n_k)$ is especially relevant for such sentences, as the contribution to E from the smaller blocks is modest.

For the record, the quantifier elimination method in [17] produces a quantifier-free formula of the form

$$\bigvee_{i=1}^I \bigwedge_{j=1}^{J_i} (h_{ij}(z) \Delta_{ij} 0)$$

where $I \leq (md)^E$, $J_i \leq (md)^{E/n_0}$, $E = \prod_{k=0}^{\omega} O(n_k)$, the degree of h_{ij} is at most $(md)^{E/n_0}$ and the Δ_{ij} are standard relations ($\geq, >, =, \neq, <, \leq$). If the coefficients of the original formula are integers of bit length at most L , the coefficients of the polynomials h_{ij} will be integers of bit length at most $(L + n_0)(md)^{E/n_0}$.

Results of Weispfenning [20], and Davenport and Heintz [9], show that the double exponential dependence on ω of the above bound on the degrees of the polynomials h_{ij} cannot be improved in the worst case.

Fisher and Rabin [10] proved an exponential worst-case lower bound for decision methods. However, the lower bound is exponential only in the number of quantifier alternations and is only singly exponential in that. A tremendous gap remains between the known upper and lower bounds for decision methods.

In closing this section we mention that work of Canny ([4], [5], [6]) has been especially influential in this area in recent years, both for the techniques he has developed and employed and for the connections he has established between the area and robotics. Work of Vorobjov [19] has also been very influential. Recent work of Pedersen [16] is especially relevant.

3 Discussion of Ideas Behind the Algorithms

3.1

We now discuss some of the main ideas behind the algorithms with the best complexity bounds. Our discussion is rather loose. At times it is not quite correct, i.e., *some of the assertions that follow are simply not quite true*. However, the discussion is *close in spirit* to a rigorous development. Only by hiding technical details can we present a comprehensible discussion that can be read in a reasonable amount of time. The reader can find all of the details in [17].

3.2

We take as our starting point a fundamental result due to Ben-Or, Kozen and Reif regarding univariate polynomials.

Let $g_1, \dots, g_m : \mathbb{R} \rightarrow \mathbb{R}$ be real univariate polynomials. To each $x \in \mathbb{R}$ there is associated a sign vector $\sigma(x) \in \mathbb{R}^m$ whose i th coordinate is defined as follows:

$$\sigma_i(x) := \begin{cases} 1 & \text{if } g_i(x) > 0, \\ 0 & \text{if } g_i(x) = 0 \\ -1 & \text{if } g_i(x) < 0. \end{cases}$$

The sign vectors of $g_1, \dots, g_m$ are the vectors $\sigma(x)$ obtained as x varies over $\mathbb{R}$. If each of the univariate polynomials g_i is of degree at most d , then the set of sign vectors has at most $2md + 1$ elements because there are at most md distinct roots and $md + 1$ open intervals defined by them.

The entire set of sign vectors of $g_1, \dots, g_m$ can be constructed efficiently by algorithms based on Sturm sequence computations. In fact, they can be constructed quickly in parallel (i.e., in polylogarithmic time), as was shown by Ken-Or, Kozen and Reif [1]. We take this fundamental result as a “given” and do not elaborate except to mention that the ideas developed in [1] can be found throughout the real decision method literature that followed its appearance. We refer to the algorithm of Ben-Or, Kozen and Reif as the BKR algorithm.

A simple corollary is the fact that there exists an efficient decision method for sentences with exactly one variable (and hence one quantifier), i.e., an efficient decision method for univariate sentences in the *existential theory of the reals*. This is because it is trivial to determine if a sentence with one quantifier is true if one knows the sign vectors of the atomic predicate polynomials.

3.3

Our first step in designing general decision and quantifier elimination methods is to provide a method for efficiently computing all sign vectors of multivariate polynomials. The approach is to efficiently reduce the set of multivariate polynomials whose sign vectors are to be determined to a set of univariate polynomials with the same sign vectors; then the BKR algorithm can be applied.

There are various ways to efficiently reduce multi-variate polynomials to appropriate univariate ones. One approach is based on the effective Nullstellensatz (Brownawell [3]). Such an approach was used by Heintz, Roy and

Solernó [14]. Another approach is based on so-called *u-resultants*. This is the approach that we discuss; it has led to the best complexity bounds.

For the moment we set aside the problem of efficiently computing the sign vectors for a set of multivariate polynomials and present a brief, but not quite accurate, discussion of *u-resultants*. Renegar [17] presents a completely accurate, but lengthy, discussion.

3.4

The *u-resultant* is an algebraic construction that allows one to reduce the problem of finding the zeros of a system of multivariate polynomials to the problem of finding the zeros of a univariate polynomial.

Let $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be a system of n polynomials in n variables and let $F : \mathbb{R}^{n+1} \rightarrow \mathbb{R}^n$ denote the homogenization of f , i.e., the terms of the i th coordinate polynomial F_i of F are obtained by multiplying the terms of f_i by the appropriate power of an additional variable x_{n+1} so as to make all terms have the same degree as that of the highest degree term in f_i . Then $F^{-1}(0)$, the zero set of F , consists of lines through the origin.

Assume that $F^{-1}(0)$ consists of only finitely many such *zero lines*. If one knows all of these zero lines, then it is a trivial matter to compute all of the zeros of f . Thus, the problem of finding the zeros of f reduces to the problem of finding the zero lines of F .

From the coefficients of F , one can construct the *u-resultant* $R : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ of F . This is a single homogeneous polynomial with the same domain as F , i.e., $\mathbb{R}^{n+1}$. If F has infinitely many zero lines, then the *u-resultant* is identically zero. However, if F has only finitely many zero lines then the zero set of the *u-resultant*, $R^{-1}(0)$, consists of those hyperplanes in $\mathbb{R}^{n+1}$ that contain the origin and are orthogonal to the zero lines of F ; so each zero line is associated with a hyperplane in $R^{-1}(0)$. It is this property that allows one to reduce the problem of finding the zero lines of F to the problem of finding the zeros of a univariate polynomial, as we now show.

Consider an arbitrary point x (other than the origin) on one of the hyperplanes of $R^{-1}(0)$. From calculus, the gradient of R at x , $\nabla R(x)$, is perpendicular to the hyperplane and, hence, lies in a zero line of F . Thus, if we could merely get a point on each of the hyperplanes of $R^{-1}(0)$, and then compute the gradient of R at those points, we would have all of the zero

lines of F . (We are glossing over subtleties like hyperplanes occurring with multiplicities, in which case the gradient will be the zero vector.)

The fact that the hyperplanes are n -dimensional makes finding points on them relatively easier than finding one-dimensional zero lines of F directly.

Consider a line $\{t\alpha + \beta; t \in \mathbb{R}\}$ that probes $\mathbb{R}^{n+1}$ in search of hyperplanes; here, α and β are fixed vectors in $\mathbb{R}^{n+1}$. Now consider the restriction of the u -resultant to this line, that is, consider the univariate polynomial $t \rightarrow R(t\alpha + \beta)$. Then t' is a zero of this univariate polynomial iff $t'\alpha + \beta$ is a point on one of the hyperplanes. Assuming, for simplicity, that the probe line is in general position relative to the hyperplanes, so that it intersects each of the hyperplanes, there is thus a corresponding zero of the univariate polynomial for each of the hyperplanes. (In a complete development one needs a method of choosing α and β to guarantee that the probe line is indeed in general position.)

We have reduced the problem of finding all of the zero lines of F to the problem of finding the zeros of a univariate polynomial, assuming F has only finitely many zero lines. In short, letting

$$\begin{aligned} p(t) &:= R(t\alpha + \beta)p : \mathbb{R} \rightarrow \mathbb{R}, \\ q(t) &:= \nabla R(t\alpha + \beta)q : \mathbb{R} \rightarrow \mathbb{R}^{n+1}, \end{aligned}$$

where α and β are fixed vectors in $\mathbb{R}^{n+1}$, we have that $p(t') = 0$ if and only if $q(t')$ is a zero line of F .

Of course one can carry out the construction of the polynomials p and q efficiently if one can construct the u -resultant efficiently. Constructing the u -resultant essentially amounts to computing the determinants of matrices. As it is well known how to compute determinants efficiently, the polynomials p and q can be computed efficiently. Again, we refer to [17] for details.

The u -resultant has been around for a long time, e.g., one can find it discussed in earlier editions of Van der Waerden [19] in the chapter on elimination theory. (Unfortunately, that chapter was eliminated in later editions.)

3.5

Now we simplify again, but not in a serious way. For our interests, the last section can be summarized as follows: $p(t') = 0$ iff $q(t')$ is a zero line of

F , where F is the homogenization of $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Rather than this, we will pretend the last section can be summarized in another way: $p(t') = 0$ iff $q(t')$ is a zero of f . In other words, we pretend the polynomials p and q given us the zeros of f directly rather than give us something that has to be dehomogenized. Our intent here is simply to make the following discussion easier to follow. We want to present the discussion in terms of the geometry of the atomic predicate polynomials occurring in a sentence or formula rather than in terms of the geometry of their homogenizations, but by doing so we are not quite honest (although we are close to being honest).

3.6

Now we show the relevance of u -resultants for the problem of computing the sign vectors of a set of multi-variate polynomials. We begin by considering a more structured problem, namely, given a polynomial system $f : \mathbb{R}^n \rightarrow \mathbb{R}$ with only finitely many zeros, and given polynomials $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$, compute the sign vectors of $g_1, \dots, g_m$ which occur at the zeros of f .

The more structured problem is easily solved. Letting $p(t)$ and $q(t)$ be as in the last subsection, we can apply the BKR algorithm to compute all sign vectors of the univariate polynomials

$$p(t), g_1(q(t)), \dots, g_m(q(t)).$$

We can then search through the sign vectors and find those whose first coordinate is zero. Clearly, the remaining coordinates of such a sign vector yield a sign vector for $g_1, \dots, g_m$ at a zero of f ; moreover, all sign vectors of $g_1, \dots, g_m$ at zeros of f are obtained in this way.

This motivates an approach to computing all sign vectors of $g_1, \dots, g_m$. First construct systems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ at whose zeros all of the sign vectors of $g_1, \dots, g_m$ occur, i.e., for each sign vector at least one of the systems has a zero at which that is indeed the sign vector. Then proceed as above.

3.7

We are now motivated to consider the following problem: Given polynomials $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$ construct systems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ at whose zeros all of

the sign vectors occur. We require each of the systems to have only finitely many zeros; otherwise, the polynomials $p(t)$ and $q(t)$ will be identically zero as the u -resultant will be identically zero; if $p(t)$ and $q(t)$ are identically zero, the preceding approach founders.

The polynomials g_i partition $\mathbb{R}^n$ into maximal connected components where two points are in the same connected component only if they have the same sign vector. This is shown schematically in Figure 1. The zero sets of the polynomials determine the boundaries of the connected components; in the figure the components are 0-, 1- and 2-dimensional. We refer to the partition as the *connected sign partition* for $g_1, \dots, g_m$.

To construct systems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ at whose zeros all of the sign vectors occur, it certainly suffices to construct systems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ which have zeros in all of the connected components.

We again change focus slightly. Consider the following problem: Given polynomials $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$ construct systems $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ with zeros in the *closures* of all of the connected components. As is indicated in section 3.8, for our purposes it actually suffices to solve this apparently easier problem. More specifically, given arbitrary polynomials $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$, it suffices to be able to compute non-vanishing pairs (p, q) , $p : \mathbb{R} \rightarrow \mathbb{R}$, $q : \mathbb{R} \rightarrow \mathbb{R}^n$ with the property that for each component of the connected sign partition for $g_1, \dots, g_m$, there exists t' for which one of the pairs (p, q) satisfies $p(t') = 0$ and $q(t')$ is in the closure of the component.

(The following discussion is somewhat technical and represents the hardest parts of [17]; the reader may wish to skip ahead to section 3.8. For the reader who does skip ahead we remark that in the following discussion we construct pairs (p, q) with the property stated in the last sentence of the previous paragraph. The construction naturally leads to the following indexing of the pairs: $(\bar{p}_A, \bar{q}_A)$ where $A \subseteq \{1, \dots, m\}$ and $\#A \leq n$, i.e., the number of elements in A does not exceed n .)

A naive solution to the problem of constructing systems with zeros in all of the closures can be obtained via calculus. Noting that the closure of each connected component has a point closest to the origin (i.e., a minimizer of $\sum_j x_j^2$ restricted to the closure) standard arguments regarding necessary conditions for optimality lead to the following: For each connected component there exists a subset $A \subseteq \{1, \dots, m\}$ such that the system $f_A : \mathbb{R}^n \rightarrow \mathbb{R}^n$

defined by

$$f_A(x) := \nabla[\det(M_A^T(x)M_A(x)) + \sum_{i \in A} (g_i(x))^2] \quad (3.7.1)$$

has a zero in the closure of the component; ∇ denotes *the gradient of* and $M_A(x)$ is the matrix whose rows are the polynomials $\nabla g_i(x)$, for $i \in A$, and the identity polynomial $x \mapsto x$. (We do not digress to discuss necessary conditions for optimality.)

There are undoubtedly numerous elementary ways to construct systems which have zeros in the closures of each of the connected components, each way having shortcomings for our purposes. The rather strange looking systems (3.7.1) we focus on have some especially useful properties. We motivate discussion of these useful properties by discussing the systems' shortcomings.

One glaring shortcoming of the systems f_A is simply that there are too many of them, i.e., 2^m . If for each system f_A we formed the corresponding polynomials $p_A(t)$ and $q_A(t)$ and applied the BKR algorithm to the univariate polynomials $g_1(q_A(t)), \dots, g_m(q_A(t))$, we would end up with an algorithm for the existential theory of the reals requiring at least 2^m operations. However, the number of atomic predicate polynomials does not appear exponentially in the best complexity bounds for decision methods, so use of all of the systems f_A is certainly not the way to proceed.

Another shortcoming of the systems f_A is that for some polynomials $g_1, \dots, g_m$ they may have infinitely many zeros. Consequently, the corresponding u -resultants, and hence the corresponding polynomials p_A and q_A , may be identically zero, in which case we get no information from the sign vectors of the univariate polynomials $g_1(q_A(t)), \dots, g_m(q_A(t))$, i.e., our approach flounders.

A way to circumvent these shortcomings is to slightly alter our approach to computing all of the sign vectors of $g_1, \dots, g_m$, the alteration depending heavily on special properties of u -resultants and the systems f_A . Briefly, as these ideas will be developed more fully in the following paragraphs, we carefully perturb the polynomials $g_1, \dots, g_m$, construct the exactly analogous systems f_A for the perturbed polynomials, construct the pairs (p_A, q_A) from these, and from these construct pairs $(\bar{p}_A, \bar{q}_A)$ which are something like limits of the pairs (p_A, q_A) as the perturbation goes to zero; the true limits of the pairs (p_A, q_A) may be identically zero, but the pairs $(\bar{p}_A, \bar{q}_A)$ will not be. The *limit* pairs $(\bar{p}_A, \bar{q}_A)$ have the property that for each connected component of the sign partition of $g_1, \dots, g_m$, there exists $A \subseteq \{1, \dots, m\}$,

$\#A \leq n$, such that for some t' , $\bar{p}_A(t') = 0$ and $\bar{q}_A(t')$ is in the closure of the connected component. Note that this circumvents both of the previously mentioned shortcomings! Even though the systems f_A constructed directly from $g_1, \dots, g_m$ may have infinitely many zeros so that we cannot construct useful pairs (p_A, q_A) from their u -resultants, we are still able to construct useful pairs indirectly! Moreover, rather than having 2^m pairs to contend with, we have $\binom{m}{n} = m^{O(n)}$ pairs to contend with, a major improvement.

Now we elaborate a bit. The perturbed polynomials we use are defined as follows:

$$g_i(x, \delta) := (1 - \delta)g_i(x) + \delta \left(1 + \sum_j i^j x_j^d \right)$$

where d is the maximal degree occurring among the unperturbed polynomials; the additional variable δ is a perturbation parameter; when $\delta = 0$ we have $x \mapsto g_i(x, \delta) = g_i(x)$.

Viewing δ as fixed consider the polynomials

$$x \mapsto g_1(x, \delta), \dots, x \mapsto g_m(x, \delta).$$

Just as we defined the systems $f_A(x)$ from the polynomials $g_1, \dots, g_m$, we can define the systems $x \mapsto f_A(x, \delta)$; simply substitute $g_i(x, \delta)$ for $g_i(x)$ in (3.7.1), the gradients referring to derivatives only in the x -variables.

Continuing to view δ as fixed, we construct the pairs of polynomials $t \mapsto p_A(t, \delta)$, $t \mapsto q_A(t, \delta)$ from the u -resultants of the systems $x \mapsto f_A(x, \delta)$. The careful choice of the perturbed polynomials and the definition of the systems $x \mapsto f_A(x, \delta)$ allow one to prove (something similar to) the following: for each sufficiently small $\delta > 0$ and for each component of the connected sign partition of $x \mapsto g_1(x, \delta), \dots, x \mapsto g_m(x, \delta)$ there exists $A \subseteq \{1, \dots, m\}$, $\#A \leq n$, and there exists $t' \in \mathbb{R}$ such that $p_A(t') = 0$ and $q_A(t')$ is in the closure of the connected component.

We would like the same conclusion to be true for $\delta = 0$. However, it may not be true for $\delta = 0$ because the systems $x \mapsto f_A(x, 0)$ may have infinitely many zeros causing the corresponding pairs $t \mapsto p_A(t, 0)$, $t \mapsto q_A(t, 0)$ to be identically zero. This problem can be handled as follows.

Tracing through definitions (all of which can be found in [17]) one finds the pairs $p_A(t, \delta)$, $q_A(t, \delta)$ to be polynomials in δ as well as in t . Expanding in powers of δ ,

$$p_A(t, \delta) = \sum_i \delta^i p_A^{[i]}(t) \quad (3.7.2)$$

$$q_A(t, \delta) = \sum_i \delta^i q_A^{[i]}(t), \quad (3.7.3)$$

let $\bar{p}_A(t) := p_A^{[i']}(t)$ where the polynomial $p_A^{[i]}(t)$ is identically zero for all $i < i'$, but not for i' ; similarly, let $\bar{q}_A(t) := q_A^{[i'']}(t)$ where the polynomial $q_A^{[i]}(t)$ is identically zero for all $i < i''$, but not for i'' .

Relying on the definition of the perturbed polynomials, now considering the perturbation parameter $\delta > 0$ to tend to zero, one can prove that (something very similar to) the following is true: For each component of the sign partition of the original polynomials $g_1, \dots, g_m$ there exists $A \subseteq \{1, \dots, m\}$, $\#A \leq n$, and there exists t' such that $\bar{p}_A(t') = 0$ and $\bar{q}_A(t')$ is in the closure of the connected component.

The idea of combining u -resultants and perturbations can be found in Canny [6], which focuses on the problem of solving degenerate systems of polynomial equations $f(x) = 0$ where $f : \mathbb{C}^n \rightarrow \mathbb{C}^n$.

3.8

Recall that our real goal is to construct all of the sign vectors of $g_1, \dots, g_m$. If in the conclusion of the next-to-last paragraph the words “the closure of” did not appear we would be in a position to accomplish the goal: simply apply the BKR algorithm to the univariate polynomials $t \mapsto g_1(\bar{q}_A(t)), \dots, t \mapsto g_m(\bar{q}_A(t))$ for all $A \subseteq \{1, \dots, m\}$ satisfying $\#A \leq n$.

However, the words “the closure of” certainly do appear, but this problem is easily remedied. From the polynomials $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$ one can easily construct polynomials $\tilde{g}_1, \dots, \tilde{g}_m : \mathbb{R}^{n+1} \rightarrow \mathbb{R}$ with the property that for each connected component C_1 in the sign partition of $g_1, \dots, g_m$ there exists a connected component C_2 in the sign partition of $\tilde{g}_1, \dots, \tilde{g}_m$ such that C_1 is the projection of the closure of C_2 onto $\mathbb{R}^n$. (We do not provide details here but, as an example, note that the positive real numbers are the projection of the closed semi-algebraic set $\{(x, y); xy \geq 1, x \geq 0, y \geq 0\}$.) Consequently, if we construct the analogous pairs $\bar{p}_A : \mathbb{R} \rightarrow \mathbb{R}, \bar{q}_A : \mathbb{R} \rightarrow \mathbb{R}^{n+1}$ for $\tilde{g}_1, \dots, \tilde{g}_m$, and discard the last coordinate polynomial of $\bar{q}_A$, we obtain appropriate pairs for $g_1, \dots, g_m$. Henceforth, we use the notation $(\bar{p}_A, \bar{q}_A)$ to denote pairs for

$g_1, \dots, g_m$ obtained in this way, so we no longer need to refer to the closure of the components.

3.9

The pairs $(\bar{p}_A, \bar{q}_A)$ constructed for $g_1, \dots, g_m$ have another property to be highlighted; the coefficients of the pairs are polynomials in the coefficients of $g_1, \dots, g_m$. If one fixes m, n , and d and considers the vector space of tuples $g_1, \dots, g_m$ of m polynomials $g_i : \mathbb{R}^n \rightarrow \mathbb{R}$ of degree at most d (identifying polynomials with their coefficients), one can replace the univariate polynomial pairs $(\bar{p}_A(t), \bar{q}_A(t))$ for fixed $g_1, \dots, g_m$, with pairs $(\bar{p}_A(g_1, \dots, g_m, t), q_A(g_1, \dots, g_m, t))$ for variable $g_1, \dots, g_m$; to obtain the corresponding univariate polynomials for particular $g_1, \dots, g_m$ just substitute the coefficients of $g_1, \dots, g_m$. In particular, *construction of the pairs can be accomplished by straight line programs*, with input being the coefficients of $g_1, \dots, g_m$. This is very important in obtaining the best complexity bounds.

(To be more accurate, to obtain the pairs as straight line programs one needs to include all pairs $(p_A^{[i]}(t), q_A^{[i]}(t))$ where $p_A^{[i]}$ and $q_A^{[i]}$ are as in (3.7.2) and (3.7.3) rather than simply $\bar{p}_A = p_A^{[i]}$ and $\bar{q}_A = q_A^{[i]}$ although for simplicity we ignore this in what follows.)

Something similar to the above pairs are constructed in Heintz, Roy and Solernó [14] by using the effective Nullstellensatz rather than u -resultants. However, the construction there is not by straight line programs.

3.10

To simplify what follows we consider the following to be established: we can construct a single pair (rather than a reasonably small set of pairs) of polynomials $p(g_1, \dots, g_m, t)$, $q(g_1, \dots, g_m, t)$ with the property that for any fixed $g_1, \dots, g_m : \mathbb{R}^n \rightarrow \mathbb{R}$ of degree at most d and any component of the connected sign partition of $g_1, \dots, g_m$ there exists t' such that $p(g_1, \dots, g_m, t') = 0$ and $q(g_1, \dots, g_m, t')$ is in the component.

Whenever we write $p(t), q(t)$, we are viewing $g_1, \dots, g_m$ as fixed as will be clear from context.

3.11

We now have an efficient decision method for the existential theory of the reals, i.e., a method for deciding sentences with only one quantifier. Simply apply the BKR algorithm to the univariate polynomials

$$g_1(q(t)), \dots, g_m(q(t))$$

where $g_1, \dots, g_m$ are the atomic predicate polynomials in the sentence, thus constructing all sign vectors for $g_1, \dots, g_m$. From these, the truth or falsity of the sentence is easily determined.

3.12

We are now in position to describe a quantifier elimination method which works inductively.

First, assume we wish to obtain a quantifier-free formula equivalent to a formula $(\exists x)P(z, x)$ with a single quantifier; here, z and x are vectors of variables, z being the free variables. The variables x can be viewed as the coefficients of the atomic predicate polynomials; different values for x yield different sentences. Viewing z as fixed, thereby obtaining a sentence with a single quantifier, we could apply the decision method of section 3.11 to determine if the sentence is true.

From this viewpoint the decision method for the existential theory of the reals is a method for deciding if sentences corresponding to different values of z are true; input to the algorithm is simply specific values for z . *Unrolling* the algorithm, which requires a close examination of the BKR algorithm, one obtains an algebraic decision tree with input z as depicted in Figure 2.

Each node of the tree is associated with a polynomial in z . Upon input z , the polynomial at the initial node is evaluated, and the value is compared to zero. Depending on the comparison, one branches to the left or to the right, thereby determining the next node and polynomial evaluation to be made, and so on until a leaf is reached.

Each leaf is assigned the value *True* or *False*. All inputs z arriving at a leaf marked *True*, correspond to sentences which are true. Similarly, all inputs z arriving at a leaf marked *False*, correspond to sentences that are false.

Here is a quantifier elimination method for the sentence: First, unroll the decision method to obtain an algebraic decision tree with input z . Next, determine all paths from the root that lead to leaves marked True. For each of these paths p , the decision method follows that path if and only if the corresponding sequence of polynomial inequalities and equalities are satisfied, i.e., z must satisfy the appropriate conjunction of polynomial equalities and inequalities

$$\bigwedge_{i \in I_p} (h_i(z) \Delta_i 0),$$

where I_p is an index set for path p and the Δ_i are standard relations ($\geq, >, =, \neq, <, \leq$). Taking the disjunction over all paths p leading to leaves marked True, one obtains a quantifier-free formula

$$\bigvee_p \bigwedge_{i \in I_p} (h_i(z) \Delta_i 0),$$

which is equivalent to $(\exists x)P(z, x)$.

3.13

The problem with this quantifier elimination method for formulae with a single quantifier is that there are too many paths in the tree; naively unrolling the decision method with input z leads to a bad complexity bound. However, this can be avoided with close examination of the BKR algorithm.

One can prove that most paths in the decision tree are not followed for any input z , although which paths are not followed is a function of the particular formula.

Rather than unroll the decision method naively, we can use the decision method to unroll itself. First, assuming the polynomial associated with the root node is $h_0(z)$ and one branches to the left if and only if $h_0(0) < 0$, we determine if the sentence $(\exists z)(h_0(z) < 0)$ is true. If it is false we know there is no need to investigate paths beginning with the edge to the left of the initial node and so we can *prune* it (and all paths containing it). Continuing in the obvious manner, pruning as we unroll the tree, we do indeed obtain a fairly efficient quantifier elimination method for formulae with a single quantifier.

3.14

To obtain a general quantifier elimination method we can simply proceed inductively on the blocks of variables in a formula

$$(Q_1x_1)\dots(Q_\omega x_\omega)P(z, x_1, \dots, x_\omega).$$

We first eliminate the block x_ω from the formula

$$(Q_\omega x_\omega)P(z, x_1, \dots, x_\omega)$$

to obtain an equivalent quantifier-free formula $P'(z, x_1, \dots, x_{\omega-1})$. We then eliminate the block $x_{\omega-1}$ from the formula

$$(Q_{\omega-1}x_{\omega-1})P'(z, x_1, \dots, x_{\omega-1})$$

to obtain an equivalent quantifier-free formula $P''(z_1, x_1, \dots, x_{\omega-2})$, and so on, until we have finally eliminated all of the quantified variables in the original formula.

This quantifier elimination method is fairly efficient. In terms of the notation introduced in section 2 it readily yields a complexity bound with $E = O(n)^{O(\omega)}$ where n is the total number of variables in the formula. This is the bound obtained by Heintz, Roy and Solernó [14] by a somewhat different approach relying on the effective Nullstellensatz rather than u -resultants.

In the following sections we describe further ideas that lead to a method with the best complexity bound, i.e., $E = \Pi O(n_k)$. The speed-up is strongly dependent on the fact that the pairs (p, q) can be computed by straight-line programs as we discussed earlier.

3.15

Our decision method for the existential theory of the reals was essentially obtained by replacing the block of variables by a single variable, that is, the sentence $(\exists x)P(x)$ with atomic predicate polynomials $g_1(x), \dots, g_m(x)$ was essentially reduced to the equivalent univariate sentence $(\exists t)P(t)$ obtained by substituting $g_i(q(t))$ for $g_i(x)$.

It is natural to attempt to extend this idea to sentences involving many quantifiers, reducing each block of variables to a single variable. We discuss a way to do this for sentences involving two quantifiers. The generalization

to arbitrary sentences if fairly straightforward; complete details can be found in [17].

Consider a sentence $(\exists y)(\forall x)P(y, x)$ with atomic predicate polynomials $g_1(y, x), \dots, g_m(y, x)$. To each fixed y there corresponds the set of sign vectors of the polynomials in x ,

$$x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x).$$

Varying y we obtain a family of sets of sign vectors.

It is easy to see that if we knew the entire family then we could decide if the sentence is true. With this as motivation, we proceed.

3.16

Consider the special case of a sentence with two quantifiers $(\exists y)(\forall x)P(y, x)$ for which x consists of a single variable; to distinguish this special case we write $(\exists y)(\forall t)P(y, t)$. Momentarily we reduce a general sentence with two quantifiers to the special case.

For the special case, to each y there corresponds the sign vectors of the univariate polynomials $t \mapsto g_1(y, t), \dots, t \mapsto g_m(y, t)$. We aim to construct the family of all such sets of sign vectors obtained as y varies over its domain.

The polynomials $g_1, \dots, g_m$ partition (y, t) -space as is shown schematically in Figure 3, i.e., the connected sign partition of $g_1, \dots, g_m$. When we focused on sentences with a single quantifier, we aimed at obtaining a point in each of the connected components of the partition. Now our goal is different.

For fixed y , the univariate polynomials $t \mapsto g_1(y, t), \dots, t \mapsto g_m(y, t)$ are simply the restrictions of the polynomials $g_1, \dots, g_m$ to the line through $(y, 0)$ that is parallel to the t -axis. It is easily seen that the sign vectors of the univariate polynomials are completely determined by which components of the connected sign partition of $g_1, \dots, g_m$ the line passes through. Now viewing y as varying continuously, the set of sign vectors of the univariate polynomials $t \mapsto g_1(y, t), \dots, t \mapsto g_m(y, t)$ can change only when y passes through one of the points indicated in Figure 3.

We thus see that y -space is partitioned into maximal connected components with the property that if y' and y'' are in the same component, then the set of sign vectors for the univariate polynomials $t \mapsto g_1(y', t), \dots, t \mapsto g_m(y', t)$ is the same as for the univariate polynomials $t \mapsto g_1(y'', t), \dots, t \mapsto g_m(y'', t)$.

$g_m(y'', t)$. Hence, if for each component of this partition of y -space we could construct a point y' in it, then we could apply the BKR algorithm to the resulting sets of univariate polynomials $t \mapsto g_1(y', t), \dots, t \mapsto g_m(y', t)$ to compute the entire family of sets of sign vectors that we desire, and from those determine if the sentence is true.

3.17

We now construct the partition of y -space, that is, we present a method for computing polynomials $h_1(y), \dots, h_\ell(y)$ with the property that if y' and y'' are in the same component of the connected sign partition of $h_1, \dots, h_\ell$ then the set of sign vectors for the univariate polynomials $t \mapsto g_1(y, t), \dots, t \mapsto g_m(y, t)$ is the same for $y = y'$ and $y = y''$. The method figures prominently in Collins' work [7] and is very well-known. (We oversimplify the method to avoid technicalities, but our oversimplification is close in spirit to the true method.)

The breakpoints in Figure 3 fall into two categories. The first category consists of points which are the projection onto y -space of intersections of the form $\{(y, t); g_i(y, t) = 0\} \cap \{(y, t); g_j(y, t) = 0\}$ for some $i \neq j$. The points y_1, y_2 and y_4 in the figure all fall into this category.

The second category consists of points which are the projection onto y -space of intersections of the form $\{(y, t); g_i(y, t) = 0\} \cap \{(y, t); \frac{d}{dt}g_i(y, t) = 0\}$. The point y_3 in the figure falls into this category.

A point y is in the first category only if the Sylvester resultant of the univariate polynomials $t \mapsto g_i(y, t)$ and $t \mapsto g_j(y, t)$ is zero at y for some $i \neq j$; the Sylvester resultant is a polynomial in the coefficients of the univariate polynomials, which here means that it is a polynomial in y .

Similarly, a point y is in the second category only if the Sylvester resultant of the univariate polynomials $t \mapsto g_i(y, t)$ and $t \mapsto \frac{d}{dt}g_i(y, t)$ is zero at y for some i .

Collecting the Sylvester resultant polynomials, we obtain the desired polynomials $h_1(y), \dots, h_\ell(y)$ which appropriately partition y -space. (Again, to be honest, the construction of appropriate polynomials h_i is a bit more complicated.)

3.18

Polynomials analogous to the h_i can be readily obtained for general sentences $(\exists y)(\forall x)P(y, x)$ involving two quantifiers, where x is not required to consist of a single variable.

First, viewing y as the *coefficients* of the atomic predicate polynomials $g_1, \dots, g_m$, from the polynomials $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ we construct the pair $p(y, t), q(y, t)$; for each component of the connected sign partition of $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ there exists t' such that $p(y, t') = 0$ and $q(y, t')$ is in the component; the pair $p(y, t), q(y, t)$ are polynomials in y as well as in t .

Then we apply the preceding Sylvester matrix determinant computations to the univariate polynomials

$$t \mapsto g_1(y, q(y, t)), \dots, t \mapsto g_m(y, q(y, t)),$$

thereby obtaining polynomials $h_1(y), \dots, h_\ell(y)$ with the following property: If y' and y'' are in the same component of the connected sign partition of $h_1, \dots, h_\ell$, then the set of sign vectors for the polynomials $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ is the same for $y = y'$ and $y = y''$.

This is where it is crucial in designing the most efficient decision and quantifier-elimination methods that the pair $p(y, t), q(y, t)$ be constructed by straight-line program, that is, the same pair works for all y (actually, recalling we have suppressed the subscripts A , the same reasonably “small” set of pairs suffices for all y). The number ℓ of polynomials $h_1, \dots, h_\ell$ does not depend on the number of variables in y . Methods whose construction of similar pairs requires branching on the coefficients y , lead to polynomials $h_1, \dots, h_\ell$ where ℓ depends exponentially on the number of variables in y . If our ℓ did depend exponentially on the number of variables in y , the approach we are now discussing would again lead to a complexity bound with $E = O(n)^{O(\omega)}$ rather than with $E = \prod_k O(n_k)$.

3.19

For an arbitrary formula $(\exists y)(\forall y)P(y, x)$ we have now described a method for constructing $h_1, \dots, h_\ell$, which partition y -space in a useful way; if for each connected component of this partition we could construct a point y' in it, then we could apply the BKR algorithm to the resulting sets of univariate

polynomials $t \mapsto g_1(y', q(y', t)), \dots, t \mapsto g_m(y', q(y', t))$ to compute the entire family of sets of sign vectors that we desire, i.e., the family of sets of sign vectors for $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ obtained as y varies.

Motivated by this, from the polynomials $h_1, \dots, h_\ell$ we construct a corresponding pair $\bar{p}(s), \bar{q}(s)$; for each component of the connected sign partition of $h_1, \dots, h_\ell$, there exists s' such that $\bar{p}(s') = 0$ and $\bar{q}(s')$ is in the component. Moreover, we already know that for each y and each component of the connected sign partition of $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ there exists t' such that $p(y, t') = 0$ and $q(y, t')$ is in the component.

Composing these facts, we have that for each set of sign vectors for polynomials of the form

$$x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x),$$

there exists s' such that $\bar{p}(s') = 0$ and the set is identical to the set of sign vectors for the polynomials

$$x \mapsto g_1(\bar{q}(s'), x), \dots, x \mapsto g_m(\bar{q}(s'), x).$$

Moreover, for each sign vector in this set there exists t' such that $p(\bar{q}(s'), t') = 0$; and the sign vector is identical to that for the vector

$$g_1(\bar{q}(s'), q(\bar{q}(s'), t')), \dots, g_m(\bar{q}(s'), q(\bar{q}(s'), t')),$$

i.e., has the same positive, zero and negative coordinates.

3.20

In essence, we have now reduced the sentence $(\exists y)(\forall x)P(y, x)$ to the bi-variate sentence $(\exists s)(\forall t)P(s, t)$ obtained by replacing the atomic predicate polynomials $g_i(y, x)$ with the polynomials $g_i(\bar{q}(s), q(\bar{q}(s), t))$. Directly generalizing this approach one can obtain a procedure for replacing each block of variables in an arbitrary sentence by a single variable. Then simply applying the quantifier elimination method of section 3.14 to the resulting sentence in many fewer variables, one obtains a decision method with a complexity bound for which $E = \prod O(n_k)^{O(1)}$.

However, this approach is not good for quantifier elimination because the block of free variables cannot be replaced by a single variable. Consequently,

only replacing each block of quantified variables by a single variable and then applying the quantifier elimination method of section 3.14, one obtains a complexity bound with $E = (n_0)^{O(\omega)} \prod_{k=1}^{\omega} O(n_k)^{O(1)}$, where n_0 is the number of free variables. Moreover, even for the decision problem alone, I see no way of refining the above approach to obtain a bound with $E = \prod O(n_k)$, rather than with $E = \prod O(n_k)^{O(1)}$.

3.21

The key to obtaining the most efficient decision and quantifier elimination methods is to also rely on the first polynomial in the pair $(\bar{p}, \bar{q})$; note that it did not appear in the approach of the previous subsection.

The reliance is via Thom's lemma (cf. Coste and Roy [8]) which asserts that if $p(t)$ is a univariate polynomial of degree d , then no two of the sign vectors of the set of derivative polynomials $p, p', \dots, p^{(d-1)}$ for which the first coordinate is zero are identical, i.e., if $p(t') = 0 = p(t'')$ then the sign of $p^{(i)}(t')$ differs from that of $p^{(i)}(t'')$ for some i . The point of this is that the sign vector of the polynomial and its derivatives at a root provides a signature for the root that allows it to be distinguished from the other roots.

Again consider an arbitrary sentence $(\exists y)(\forall x)P(y, x)$ with two quantifiers, and let $(p, q), (\bar{p}, \bar{q})$ be as before. Consider the set of sign vectors of the system of bivariate polynomials,

$$\begin{aligned} \frac{d^i}{ds^i} \bar{p}(s) & \quad i = 0, \dots, D \\ g_i(\bar{q}(s), q(\bar{q}(s), t)) & \quad i = 1, \dots, m, \end{aligned} \tag{3.21.1}$$

where D is at least as large as the degree of $\bar{p}$. As we already know from earlier sections, the entire set of sign vectors for these polynomials can be computed efficiently.

We claim that from this set of sign vectors one can easily construct the family of sets of sign vectors for the polynomials $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ obtained as y varies. To see this first recall two observations recorded in section 3.19: (i) For each component of the connected sign partition of $h_1, \dots, h_\ell$ there exists s' such that $\bar{p}(s') = 0$ and $\bar{q}(s')$ is in the component; (ii) For each y and each component of the connected sign partition of $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ there exists t' such that $p(y, t') = 0$ and $q(y, t')$ is in the component.

Proceed as follows. First partition the set of sign vectors for the set of polynomials (3.21.1) according to the coordinates corresponding to the polynomials $\frac{d^i}{ds^i}\bar{p}(s)$, $i = 0, \dots, D$, grouping those vectors together for which the coordinates are identical and discarding all vectors for which the first of the coordinates is not zero. By Thom's lemma, for each group there exists a root s' of $\bar{p}$ such that the sign vectors in the group are precisely those occurring at points of the form (s', t) . Hence, by (i) and (ii) above, for each y there exists one of the groups such that the sign vectors of $x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$ are identical to the sign vectors obtained by truncating the sign vectors in the group, where the truncation leaves only the coordinates corresponding to the polynomials

$$g_1(\bar{q}(s), q(\bar{q}(s), t)), \dots, g_m(\bar{q}(s), q(\bar{q}(s), t)).$$

Conversely, for each of the groups the truncated sign vectors are precisely the sign vectors of

$$x \mapsto g_1(y, x), \dots, x \mapsto g_m(y, x)$$

for some y .

We thereby obtain another approach to determining if a sentence with two quantifiers is true. This approach directly reduces the general decision problem to the problem of determining signs of multi-variate polynomials, and allows us to avoid unrolling any algorithms.

3.22

The above procedure generalizes in a straightforward manner to arbitrary sentences

$$(Q_1 x_1) \dots (Q_\omega x_\omega) P(x_1, \dots, x_\omega).$$

First viewing $(x_1, \dots, x_{\omega-1})$ as coefficients of the atomic predicate polynomials, one constructs a pair $(p^{[\omega-1]}, q^{[\omega-1]})$ and polynomials $h_i^{[\omega-1]}$ which partition $(x_1, \dots, x_{\omega-1})$ -space. Then, viewing $(x_1, \dots, x_{\omega-2})$ as coefficients of the polynomials $h_i^{[\omega-1]}$, one constructs a pair $(p^{[\omega-2]}, q^{[\omega-2]})$ and polynomials $h_i^{[\omega-2]}$, which partition $(x_1, \dots, x_{\omega-2})$ -space. And so on.

Next, one computes the sign vectors of the polynomials

$$\begin{aligned}
& \frac{d_i}{dt_1^i} p^{[1]}(t_1) \quad i = 0, \dots, D \\
& \frac{d_i}{dt_2^i} p^{[2]}(q^{[1]}(t_1), t_2) \quad i = 0, \dots, D \\
& \vdots \\
& \frac{d_i}{dt_{\omega-1}^i} p^{[\omega-1]}(q^{[1]}(t_1), q^{[2]}(q^{[1]}(t_1), t_2), \dots, \\
& \quad q^{[\omega-2]}(q^{[1]}(t_1), q^{[2]}(q^{[1]}(t_1), t_2), \dots, t_{\omega-2}), t_{\omega-1}) \quad i = 0, \dots, D \\
& g_i(q^{[1]}(t_1), q^{[2]}(q^{[1]}(t_1), t_2), \dots, \\
& \quad q^{[\omega-1]}(q^{[1]}(t_1), q^{[2]}(q^{[1]}(t_1), t_2), \dots, t_{\omega-1}), t_{\omega})) \quad i = 1, \dots, m.
\end{aligned}$$

These are the polynomials obtained by first substituting $q^{[\omega]}(x_1, \dots, x_{\omega-1}, t_{\omega})$ for x_{ω} in the polynomials $(d^i/dt_j)p(x_1, \dots, x_{j-1}, t_j)$ and $g_i(x_1, \dots, x_{\omega})$. Then, one substitutes $q^{[\omega-1]}(x_1, \dots, x_{\omega-2}, t_{\omega-2})$ for $x_{\omega-2}$ in the resulting polynomials, and so on. From these sign vectors one is able to determine if the original sentence is true. (See [17] for the details.)

3.23

The principal advantage to the above decision method in contrast to the one obtained in section 3.20 by replacing each block of variables with a single variable and then applying the quantifier elimination method of section 3.14 is that the above method does not require any algorithm to be unrolled into an algebraic decision tree; the quantifier elimination method of section 3.14 was obtained by repeated unrolling, once each time a block of quantified variables was eliminated.

To obtain a quantifier elimination method for an arbitrary formula

$$(Q_1 x_1) \dots (Q_{\omega} x_{\omega}) P(z, x_1, \dots, x_{\omega})$$

with free variables z we can unroll our new decision method into an algebraic decision tree in z , *pruning* branches (as in section 3.13) as we unroll. Hence, to obtain a quantifier elimination method, we need only unroll once as opposed to once for each block of quantified variables. In this manner one is led to a quantifier elimination method with complexity bound for which $E = \prod_{k=0}^{\omega} O(n_k)^{O(1)}$. With considerable attention to detail, the method can be refined to provide a complexity bound for which $E = \prod O(n_k)$, the best available; once again, we remark that the refinements are in [17].

References

- [1] Ben-Or, M., Kozen, D. and Reif, J., *The complexity of elementary algebra and geometry*. J. Comput. System Sci. **32** (1986), 251-264.
- [2] Blum, L., Shub, M. and Smale, S., *On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines*. Bull. Amer. Math. Soc. (N.S.) **21** (1989), 1-46.
- [3] Brownawell, D., *Bounds for the degrees of the Nullstellensatz*. Ann. of Math. (2) **126** (1987), 577-591.
- [4] Canny, J., *The complexity of robot motion planning*. MIT Press, Cambridge, MA, 1989.
- [5] Canny, J., *Some algebraic and geometric computations in PSPACE*. Proc. 20th Annual ACM Symp. on the Theory of Computing, Association for Computing Machinery, NY, 1988, pp. 460-467.
- [6] Canny, J., *Generalized characteristic polynomials*, J. Symbolic Comput. **9** (1990), 241-250 .
- [7] Collins, G., *Quantifier elimination for real closed fields by cylindrical algebraic decomposition*, Lectures Notes in Comput. Sci., vol. 33, Springer-Verlag, New York, Berlin and Heidelberg, 1975, pp. 515-532.
- [8] Coste, M. and Roy, M.F., *Thom's lemma, the coding of real algebraic numbers and the computation of the topology of semi-algebraic sets*. J. Symbolic Comput. **5** (1988), 121-129.
- [9] Davenport, J. and Heintz, J., *Real quantifier elimination is doubly exponential*. J. Symbolic Comput. **5** (1988), 29-35 .
- [10] Fischer, M. and Rabin, M., *Super-exponential complexity of Presburger arithmetic*, *Complexity of Computations*, SIAM-AMS Proc., vol. 7), Amer. Math. Soc., Providence, RI, 1974, pp. 27-41.
- [11] Fitchas, N., Galligo, A. and Morgenstern, J., *Algorithmes rapides en sequentiel et en parallel pour l'e limination de quantificateurs en geometrie e le mtaire. Se minaire Structures Ordonnees*, U.E.R. de Math. Univ. Paris VII, 1987.

- [12] Grigor'ev, D. Yu., *The complexity of deciding Tarski algebra*. J. Symbolic Comput. **5** (1988), 65-108.
- [13] Grigor'ev, D. Yu. and Vorobjov, N.N. Jr., *Solving systems of polynomial inequalities in subexponential time*. J. Symbolic Comput. **5** (1988), 37-64.
- [14] Heintz, J., Roy, M.-F. and Solernó, P., *Sur la complexite du principe de Tarski-Seidenberg*. Bull. Soc. Math. France **118** (1990), 101-126.
- [15] Neff, A.C., *Specified precision polynomial root isolation is in NC*. J. Comput. Systems Sci. (to appear).
- [16] Pedersen, P., *Counting real zeros*. Ph.D. Thesis, Courant Inst. of Math. Sci., New York Univ., 1991
- [17] Renegar, J., *On the computational complexity and geometry of the first-order theory of the reals*. J. Symbolic Comput. (to appear).
- [18] Tarski, A., *A decision method for elementary algebra and geometry*. Univ. of California Press, Berkeley, CA, 1951.
- [19] Van der Waerden, B.L., *Modern algebra*, Vol. II. Ungar, 1950.
- [20] Weispfenning, V., *The complexity of linear problems in fields*. J. Symbolic Comput. **5** (1988), 3-27.