THE COMPLEXITY OF ELEMENTARY ALGEBRA AND GEOMETRY
(preliminary version)

Michael Ben-Or, MIT
Dexter Kozen, IBM Research
John Reif, Harvard

In memoriam
Alfred Tarski
1902-1983

1. Introduction

In his 1948 paper "A Decision Method for Elementary Algebra and Geometry," Alfred Tarski gave a decision procedure for the first-order theory of the real numbers with +, ·, and =, commonly known as the theory of real closed fields. His decision procedure was nonelementary (in the complexity-theoretic sense). An elementary decision procedure was given by Leonard Monk in his Berkeley PhD thesis, and in 1974, double-exponential-time decision procedures were given independently by Collins and by Monk/Solovay. Various improvements and heuristics notwithstanding, that worst-case bound has stood since that time.

In this paper we show that the problem is exponential space complete. Our exponential space algorithm can also be implemented in parallel exponential time.

The main lemma is of independent interest: a $\log(n)^{O(1)}$-depth, polynomial-size circuit to determine whether a given set of rational, univariate polynomial equations and inequalities has a real solution. It is known how to solve the problem sequentially in polynomial time [C].

The practical importance of these problems cannot be overstated. Polynomial manipulation is crucial in symbolic manipulation and nonlinear optimization (see e.g. [S]). We do not expect to be able to implement our parallel algorithms in VLSI, given the current state of technology; however, the mathematical ideas behind them may give simpler sequential algorithms than the ones currently in use, notably the cylindric algebraic decomposition (CAD) method of Collins [C]. CAD relies on numerical techniques that seem to be inherently sequential. Our method is, on the other hand, purely algebraic. We return to Tarski's original formulation, and obtain parallelization by exploiting the algebraic structure to the limit, relying heavily on recent advances in parallel algorithms for matrix and polynomial algebra [Cs,BGH,vzG].

In the main lemma (testing consistency of polynomial constraints), we devise a generalized

Sturm sequence method, and give an efficient parallel implementation. The basic idea behind this method appears in Tarski's paper, but our parallel implementation requires that we develop the idea considerably further.

2. Testing consistency of univariate polynomial constraints

Suppose we are given a finite set Σ of rational univariate polynomials $p(x)$, and a sign assignment $\sigma:\Sigma \to \{-1,0,1\}$ representing the system of equations and strict inequalities

$$p(x) \begin{cases} < 0, \text{ if } \sigma(p) = -1 \\ = 0, \text{ if } \sigma(p) = 0 \\ > 0, \text{ if } \sigma(p) = 1 \end{cases}.$$

The system is said to be consistent if it has a real solution. We write $\|\Sigma\| \le n$ if there are at most n polynomials in Σ, each $p \in \Sigma$ is of degree at most n, and all coefficients of $p \in \Sigma$ can be represented with at most n bits. We will give an algorithm in NC (that is, a circuit of depth $\log(n)^{O(1)}$ and size $n^{O(1)}$) to determine whether a system (Σ,σ) with $\|\Sigma\| \le n$ is consistent. In general, we know of no NC circuit that finds a real solution of (Σ,σ), even if we know that one exists; however, we can test consistency, and even say how many solutions there are.

For the real closed field algorithm, we will need more: for a given Σ, we will need to produce list of all consistent sign assignments. Although there are exponentially many possible sign assignments, if $\|\Sigma\| \le n$ then at most $2n^2+1$ of them are consistent. We show how to do this in NC and space $\log(n)^{O(1)}$.

2.1 Simple refinement

A polynomial is called simple if it has only simple roots (multiplicity = 1). A set of polynomials Σ is called simple if the elements of Σ are simple and pairwise relatively prime. Given Σ with $\|\Sigma\| \le n$, we show how to produce in NC a simple refinement, that is, a new set of polynomials Γ such that Γ is simple, $\|\Gamma\| \le n^2\log(n)$, and each $p \in \Sigma$ is a product of powers of elements of Γ. A sign assignment for Γ will uniquely determine a sign assignment for Σ. This step is not essential, but it does simplify the presentation substantially.

A straightforward divide-and-conquer yields nonpolynomial growth in degree and coefficient size, so care must be taken. We will need to use the parallel multiple-polynomial gcd algorithm of [vzG].

First we show how to refine a single polynomial p. Let q_i be the product of all linear factors $x - \theta$ of p such that θ is a root of p of multiplicity i. Thus

$$p = q_1 q_2^2 \cdots q_n^n .$$

If θ is a root of p of multiplicity m, then θ is a root of p' of multiplicity $m - 1$. Thus for $i \le m$, θ is a root of $p^{(i)}$ of multiplicity $m - i$. In particular $x - \theta$ does not divide $p^{(m)}$. Thus

$$\gcd(p,p',\ldots,p^{(i)}) = q_{i+1} q_{i+2}^2 \cdots q_n^{n-i}$$

and

$$q_i = \frac{\gcd(p,p',\ldots,p^{(i-1)})\cdot\gcd(p,p',\ldots,p^{(i+1)})}{\gcd(p,p',\ldots,p^{(i)})^2}$$

The q_i constitute a simple refinement of p.

Thus we can assume that all roots of each $p \in \Sigma$ are simple. Let $\theta(p)$ denote the set of roots of p. Note

$$(*) \qquad \begin{aligned} \cap_{p\in I}\ \theta(p) &= \theta(\gcd(I)) \\ \cup_{p\in I}\ \theta(p) &= \theta(\mathrm{lcm}(I)) \\ \neg\theta(p) &= \theta(\mathrm{lcm}(\Sigma)/p)\ , \end{aligned}$$

where $I \subseteq \Sigma$ and $\neg$ denotes complementation in $\theta(\mathrm{lcm}(\Sigma))$, the set of all roots of all $p \in \Sigma$. Since $\|\Sigma\| \leq n$,

$$|\theta(\mathrm{lcm}(\Sigma))| \leq n^2\ .$$

The $\theta(p)$, $p \in \Sigma$, are generators of a Boolean algebra on $\theta(\mathrm{lcm}(\Sigma))$ with Boolean operations given by (*). The simple refinement we seek consists of the atoms of this Boolean algebra, i.e. all nonempty sets of the form

$$\cap_{p\in I}\ \theta(p) \cap \cap_{p\in I'}\ \neg\theta(p)$$

where $I \cup I' = \Sigma$ and $I \cap I' = \emptyset$. By (*) these atoms are represented by the polynomials

$$p(I,I') = \gcd(I)/\gcd(\gcd(I),\mathrm{lcm}(I'))$$

of nonzero degree. These will be determined in log(n) stages. At some intermediate stage, suppose we have all the atoms of the Boolean algebra generated by $\Sigma_1 \subseteq \Sigma$, given by a list of pairs (I,I') where $I \cup I' = \Sigma_1$ and $I \cap I' = \emptyset$, and all atoms of Σ_2 presented in a similar way. Each atom of $\Sigma_1 \cup \Sigma_2$ is an intersection of an atom (I,I') of Σ_1 and an atom (J,J') of Σ_2. This intersection is represented by

$$p(I \cup J, I' \cup J')\ .$$

Computing all such polynomials and throwing out those of degree 0, we are left with a list of pairs (K,K') representing all atoms of $\Sigma_1 \cup \Sigma_2$. After log(n) stages we have built a list of all atoms for Σ. The corresponding polynomials provide a simple refinement of Σ. The entire computation can be done in NC using the multiple-polynomial gcd algorithm of [vzG].

If Γ is a simple refinement of Σ, then any consistent sign assignment for Γ gives a consistent sign assignment for Σ. Any consistent sign assignment for Γ assigns at most one 0, since the elements of Γ are relatively prime. We wish to refine Γ further to get Δ such that any consistent sign assignment for Σ is obtained from a consistent sign assignment for Δ in which <u>exactly</u> one 0 is assigned. This is done by appending the polynomial

$$r = \Pi(\Sigma)'\cdot(x - b)\cdot(x + b)$$

to Σ before refining, where

$$b = 1 + \max_{0\leq i\leq k-1} |a_i/a_k|\ ,$$

the a_i are the coefficients of $\Pi(\Sigma)$, and a_k is the leading coefficient. r has a root in every interval between any two roots of elements of Σ,

and roots bounding all the roots of Σ [M]. Thus without loss of generality we can limit our attention to sign assignments that assign exactly one 0.

2.2 A generalized version of Sturm's theorem

We have reduced the problem to the following: given p and Σ, $\{p\} \cup \Sigma$ simple, $\|\{p\} \cup \Sigma\| \leq n$, list all satisfiable sign assignments assigning 0 to p and -1 or 1 to the elements of Σ.

If $\Sigma = \emptyset$, the problem is merely to determine whether p has any real roots. This can be solved using Sturm sequences [see SS]. This technique uses the coefficients of the polynomial remainder sequence for p and p', which can be obtained as subresultants, or determinants of submatrices of the Sylvester matrix of coefficients of p and p', in NC [BT].

Sturm sequences work as follows. Let $a,b \in R$, $a < b$, $p(a) \neq 0$ and $p(b) \neq 0$. Consider the Euclidean remainder sequence

$$p_0 = p$$
$$p_1 = p'$$
$$\ldots$$
$$p_{k+1} = q_k p_k - p_{k-1}$$
$$\ldots$$
$$p_n$$

where p_{k+1} is the negative of the remainder obtained by dividing p_{k-1} by p_k. p_n is a constant nonzero polynomial, since by assumption p_0, p_1 are relatively prime. Count the number of sign changes in $p_0(a),\ldots,p_n(a)$, count the number of sign changes in $p_0(b),\ldots,p_n(b)$, and subtract. Sturm's theorem states that the result is the number of real roots of p in the interval (a,b). If lt(q) is the leading term of q, then evaluating lt(q) at -1 gives the same sign as evaluating q at a number smaller than all the real roots of q; similarly, evaluating lt(q) at 1 gives the same sign as evaluating q at a number larger than all the real roots of q. Thus if we subtract the number of sign changes in $lt(p_0)(1),\ldots,lt(p_n)(1)$ from the number of sign changes in $lt(p_0)(-1),\ldots,lt(p_n)(-1)$, the result is the number of real roots of p. Denote the result of this computation by S(p,p').

Define

$$c_q = \{x \mid p(x) = 0 \text{ and } q(x) > 0\}$$
$$\bar{c}_q = \{x \mid p(x) = 0 \text{ and } q(x) < 0\}$$

where q is simple and relatively prime to p. Sturm's theorem states that

$$S(p,p') = |c_q| + |\bar{c}_q| .$$

The following lemma is implicit in Tarski's paper.

Lemma [T].

$$S(p,p'q) = |c_q| - |\bar{c}_q| .$$

Proof. We sketch the proof for the reader familiar with the proof of Sturm's theorem. There are four cases, depending on the signs of p' and q at a root of p. If $p' > 0$ and $q > 0$, then $p'q > 0$ and S(p,p'q) gains a sign change as we jump over

the root. If $p' > 0$ and $q < 0$, then $p'q < 0$ and $S(p,p'q)$ loses a sign change. If $p' < 0$ and $q > 0$, then $p'q < 0$ and we gain a sign change. If $p' < 0$ and $q < 0$, then $p'q > 0$ and we lose. Thus we gain a sign change whenever $q > 0$, and lose a sign change whenever $q < 0$, so the net gain or loss is as stated in the lemma. □

2.3 A tensor identity

If $\Sigma = \{q\}$, the values of $S(p,p')$ and $S(p,p'q)$ determine the values of $|c_q|$ and $|\bar{c}_q|$ by solving a simple nonsingular linear system of order 2:

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} |c_q| \\ |\bar{c}_q| \end{bmatrix} = \begin{bmatrix} S(p,p') \\ S(p,p'q) \end{bmatrix}$$

The 2×2 matrix is denoted A_1. For $\Sigma = \{q_1,q_2\}$, there are four linear equations in four unknowns:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} |c_1 \cap c_2| \\ |\bar{c}_1 \cap c_2| \\ |c_1 \cap \bar{c}_2| \\ |\bar{c}_1 \cap \bar{c}_2| \end{bmatrix} = \begin{bmatrix} S(p,p') \\ S(p,p'q_1) \\ S(p,p'q_2) \\ S(p,p'q_1q_2) \end{bmatrix}$$

Here c_i and $\bar{c}_i$ abbreviate c_{q_i} and $\bar{c}_{q_i}$, respectively. The matrix A_2 is a 4×4 Hadamard matrix, for which $A^{-1} = \frac{1}{4} \cdot A$. Note that it is also the Kronecker (tensor) product of A_1 with itself. After making the four Sturm queries on the right, the system can be solved for the four unknowns, which will be nonzero exactly when the corresponding sign assignment is consistent.

In general, for $\Sigma = \{q_1,\ldots,q_n\}$, we get a $2^n \times 2^n$ linear system $A_n c = s$, where c is a vector consisting of all elements of the form

$$|d_1 \cap \ldots \cap d_n|, \quad d_i \in \{c_i,\bar{c}_i\}, \ 1 \le i \le n,$$

and s is a vector of all elements of the form

$$S(p,p' \cdot \Pi(I))$$

where $I \subseteq \Sigma$. A_n is a $2^n \times 2^n$ Hadamard matrix, obtained by taking the Kronecker product of A_1 with itself n times. A_n is easily inverted (in fact, $A_n^{-1} = 2^{-n} \cdot A_n$), so we could in principle make all the Sturm queries s and solve the system for c. The nonzero elements of c correspond to the consistent sign assignments. Unfortunately, the system is far too big for this computation to be done in NC.

We now make the key observation that since $\|\{p,q_1,\ldots,q_n\}\| \le n$, p is of degree at most n, therefore all but at most n of the elements of c are 0. Thus $A_n c = s$ is equivalent to a much smaller system in which c contains only those elements that are nonzero, obtained by dropping out the zero elements of c and the corresponding columns of A_n, finding a basis among the rows of the resulting matrix, and deleting the other rows and corresponding elements of s.

The problem now is to construct the smaller system without constructing all of A_n first. We proceed by divide-and-conquer, constructing it in log(n) stages. At some intermediate stage, suppose we have constructed a system $Ac = s$ of order at most n solving the problem for $\{p\} \cup \Gamma$, and a system $A'c' = s'$ of order at most n solving the problem for $\{p\} \cup \Gamma'$, where $\Gamma,\Gamma' \subseteq \Sigma$ and $\Gamma \cap \Gamma' = \emptyset$. We wish to combine these into a system of order

n for $\{p\} \cup \Gamma \cup \Gamma'$. An intersection

$$|\cap_{q\in I} c_q \cap \cap_{q\in\Gamma-I} \bar{c}_q|$$

for some $I \subseteq \Gamma$ appears in c iff it is nonzero, and similarly for c'.

First we combine A and A' by taking their Kronecker product $A \otimes A'$, obtained by replacing each entry a_{ij} of A by the matrix $a_{ij}\cdot A'$. The matrix $A \otimes A'$ is nonsingular since A and A' are. We index the entries of $A \otimes A'$ by four indices i,j,k,l where i,j give the position of the block $a_{ij}\cdot A'$ and k,l give the position of the entry $a_{ij}\cdot a_{kl}'$ within the block.

The vectors c and c' are combined into a new column vector $c \otimes c'$ consisting of $|A|$ column vectors placed end-to-end, each one of length $|A'|$. ($|A|$ denotes the order of A.) The entries of $c \otimes c'$ are indexed by two indices i,j, where i gives the position of the block and j gives the position of the entry within the block. The i,j^{th} entry of $c \otimes c'$ is $|e \cap e'|$, where $|e|$ is the i^{th} entry of c and $|e'|$ is the j^{th} entry of c'.

The vectors s and s' are combined into a new column vector $s \otimes s'$ consisting of $|A|$ column vectors placed end-to-end, each one of length $|A'|$. The entries of $s \otimes s'$ are indexed by i,j as above. The i,j^{th} entry of $s \otimes s'$ is

$$S(p,p'\cdot\Pi(I\oplus I'))\ ,$$

where $I \subseteq \Gamma$ and $S(p,p'\cdot\Pi(I))$ is the i^{th} entry of s, $I' \subseteq \Gamma'$ and $S(p,p'\cdot\Pi(I'))$ is the j^{th} entry of s', and $\oplus$ is exclusive-or.

The following equation gives the relationship between these constructs.

Lemma. $A \otimes A'(c \otimes c') = s \otimes s'$.

Proof. We omit the proof in this version. □

All nonzero intersections of the form

$$|\cap_{q\in I} c_q \cap \cap_{q\in\Gamma\cup\Gamma'-I} \bar{c}_q|\ ,$$

$I \subseteq \Gamma \cup \Gamma'$, are contained in $c \otimes c'$, so anything not appearing there is 0. The system $A \otimes A'$ is of order $|A|\cdot|A'|$, which is never more than n^2. All but n of the entries of $c \otimes c'$, however, are 0. As above, we can compute $s \otimes s'$ and solve for $c \otimes c'$, discard the 0 entries of $c \otimes c'$ and the corresponding columns of $A \otimes A'$, find a basis among the rows of the resulting matrix, and discard the other rows and the corresponding entries of $s \otimes s'$. This gives a nonsingular system of order at most n that accounts for all consistent sign assignments of $\{p\} \cup \Gamma \cup \Gamma'$. After log(n) stages, we have a system giving all consistent sign assignments of $\{p\} \cup \Sigma$. Each stage requires the solution of a nonsingular system of order at most n^2, as well as the computation of a basis; all these computations can be done in NC [Cs,vzG,BGH].

3. Testing consistency of multivariate polynomial constraints

The construction of Section 2 produced a family of circuits, denoted generically by C, to list all

consistent sign assignments of a set Σ of polynomials, $\|\Sigma\| \le n$. It is crucial to observe that C did not need to know the actual values of the coefficients of $p \in \Sigma$, but only the signs of certain polynomials in the coefficients of p. These polynomials arose in the Sturm computations, gcd computations, subresultants, etc.

This observation allows us to construct circuits to handle multivariate polynomials. Given a set of polynomials $\Sigma(x_1,\ldots,x_k)$ in $Q[x_1,\ldots,x_k]$, we write them as polynomials in x_k with coefficients in the polynomial ring $Q[x_1,\ldots,x_{k-1}]$. In order to list the satisfiable sign assignments of $\Sigma(x_1,\ldots,x_k)$, we need only know the signs of certain polynomials $\Sigma(x_1,\ldots,x_{k-1})$ in $Q[x_1,\ldots,x_{k-1}]$. A careful complexity analysis of the construction of Section 2 reveals that

$$\|\Sigma(x_1,\ldots,x_{k-1})\| \le O(\|\Sigma(x_1,\ldots,x_k)\|^3) .$$

Suppose we have built a circuit C_{k-1} to list all consistent sign assignments of any input set of polynomials Σ over $Q[x_1,\ldots,x_{k-1}]$, $\|\Sigma\| \le c\cdot\|\Sigma(x_1,\ldots,x_k)\|^3$. If we knew the set $\Sigma(x_1,\ldots,x_{k-1})$ in advance, we could apply C_{k-1} to $\Sigma(x_1,\ldots,x_{k-1})$ to obtain its consistent sign assignments; then each consistent sign assignment σ of $\Sigma(x_1,\ldots,x_{k-1})$ would provide enough input information to the circuit C to enable it to list all consistent sign assignments of $\Sigma(x_1,\ldots,x_k)$ consistent with σ. By doing this in parallel for all such σ, we would get the circuit C_k listing all consistent sign assignments of $\Sigma(x_1,\ldots,x_k)$.

The situation is a bit more complicated than that just described, because the polynomials $\Sigma(x_1,\ldots,x_{k-1})$ are not known to C at the time of input but are generated along the way. Thus C must use C_{k-1} as a subroutine, calling it at each level to incorporate new polynomials into $\Sigma(x_1,\ldots,x_{k-1})$ as they are generated, and producing new consistent sign assignments that extend the old assignments with signs for the new polynomials.

Using the rough estimate

$$\|\Sigma(x_1,\ldots,x_{k-1})\| \lesssim \|\Sigma(x_1,\ldots,x_k)\|^3 ,$$

we get

$$\|\Sigma(x_1,\ldots,x_i)\| \lesssim \|\Sigma(x_1,\ldots,x_k)\|^{3^{k-i}} , \quad i \le k.$$

The depth of C_k is the product of the depth of C, roughly $(\log\|\Sigma(x_1,\ldots,x_{k-1})\|)^3$, and the depth of C_{k-1}. Inductively this gives

$$\begin{aligned} \text{depth}(C_k) &\lesssim \Pi_{0\le i\le k}\ (\log\|\Sigma(x_1,\ldots,x_i)\|)^3 \\ &\lesssim \Pi_{0\le i\le k}\ (\log(\|\Sigma(x_1,\ldots,x_k)\|^{3^i}))^3 \\ &\lesssim 9^{k^2}\cdot(\log\|\Sigma(x_1,\ldots,x_k)\|)^{3k} , \end{aligned}$$

which is still NC for fixed number of variables k. If the number of variables can grow linearly with the size of the input, however, then the depth becomes exponential. This is the source of the exponential upper bound in the quantifier elimination procedure that follows.

4. Elimination of quantifiers

Once we have a circuit to produce the $\Sigma(x_1,\ldots,x_k)$ and their satisfiable sign

assignments, the actual quantifier elimination circuit is straightforward. Suppose we want to decide the truth of the sentence

$$Q_1 x_1 \cdots Q_n x_n \; B(x_1,\ldots,x_n)$$

where $B(x_1,\ldots,x_n)$ is a Boolean combination of polynomial equations and inequalities in $x_1,\ldots,x_n$. Let $\Sigma(x_1,\ldots,x_n)$ be this set of polynomials, and generate the sets $\Sigma(x_1,\ldots,x_i)$, $1 \le i \le n$, as in Section 3. If $Q_1 = \exists$, construct an $\vee$-branch with root r and one leaf for each consistent sign assignment of $\Sigma(x_1)$. If $Q_1 = \forall$, the construction is the same, except we use an $\wedge$-branch instead of an $\vee$-branch. For each leaf, the consistent sign assignment σ associated with that leaf determines a set of consistent sign assignments for $\Sigma(x_1,x_2)$, namely those assignments that are consistent with σ. If $Q_2 = \exists$, we again construct an $\vee$-branch from σ, and each new leaf is associated with a consistent sign assignment of $\Sigma(x_1,x_2)$ that is consistent with σ. Continuing in this fashion, at the bottom of the circuit we have consistent sign assignments for $\Sigma(x_1,\ldots,x_n)$, which determine the truth or falsity of $B(x_1,\ldots,x_n)$. Starting from these truth values, the circuit associates a Boolean value with each node, computing upward toward the root r. The final Boolean value associated with r is <u>true</u> iff $Q_1x_1\ldots Q_nx_n$ $B(x_1,\ldots,x_n)$ is true. The circuit is not really a tree but a directed acyclic graph; the depth and size of the circuit are roughly those of the circuit C_n constructed in Section 3.

The exponential-depth circuits produced above are uniform and can be simulated in exponential space in a straightforward way.

5. A lower bound

The problem was previously known to be hard for alternating exponential time restricted to a linear number of alternations [B]. We use the techniques of [MM] to show that the triviality problem for polynomial ideals is exponential-space hard. This problem reduces easily to the theory of the complex numbers using Hilbert's nullstellensatz. The theory of the complexes reduces easily to the theory of the reals. This gives us exponential-space completeness of the theory of complexes as well as the theory of reals.

References

[B] Berman, L., "The complexity of logical theories," Theor. Comput. Sci. 11 (1980), 71-77.

[BGH] Borodin, Hopcroft, von zur Gathen, "Fast parallel matrix and gcd computations," Proc. 23rd Symp. on Foundations of Computer Science, Nov. 1982, 65-71.

[BT] Brown, W. and J. F. Traub, "On Euclid's algorithm and the theory of subresultants," J. ACM 18 (1971), 505-514.

[C] Collins, G.E., "Quantifier elimination for real closed fields by cylindric algebraic decomposition," Proc. 2nd GI Conference on Automata Theory and Formal Languages, Springer-Verlag LNCS 35, Berlin, 1975, 134-183.

[M] Marden, M. <u>Geometry of Polynomials.</u> Amer. Math. Soc., Providence, 1966.

[MM] Mayr, E.W. and A.R. Meyer, "The complexity of the word problem for commutative semigroups and polynomial ideals," Adv. in Math 46 (1982), 305-329.

[Cs] Csanky, L., "Fast parallel matrix inversion algorithms," SIAM J. Comput. 5 (1976), 618-623.

[vzG] von zur Gathen, J., "Parallel algorithms for algebraic problems," Proc. 15th ACM Symp. on Theory of Computing, April 1983, 17-23.

[M] Monk, L. "An elementary-recursive decision procedure for Th(R,+,·)," manuscript, U. C. Berkeley, 1974.

[S] Proc. ACM Symp. on Symbolic and Algebraic Computation, ed. Jenks, Yorktown Heights, NY, 1976.

[SS] Schwartz, J.T., and M. Sharir, "On the piano mover's problem II. General techniques for computing topological properties of real algebraic manifolds," Adv. in Appl. Math 4 (1983), 298-351.

[T] Tarski, A. "A decision method for elementary algebra and geometry," U. of Calif. Press, 1948; 2nd edition, 1951.