Arithmetic with real algebraic numbers is in NC

Bud Mishra Paul Pedersen
New York University

Abstract

We describe NC algorithms for doing exact arithmetic with real algebraic numbers in the sign-coded representation introduced by Coste and Roy [CoR 1988]. We present polynomial sized circuits of depth $O(\log^3 N)$ for the monadic operations $-\alpha$, $1/\alpha$, as well as $\alpha + r$, $\alpha \cdot r$, and $sgn(\alpha - r)$, where r is rational and α is real algebraic. We also present polynomial sized circuits of depth $O(\log^7 N)$ for the dyadic operations $\alpha+\beta$, $\alpha\cdot\beta$, and $sgn(\alpha-\beta)$, where α and β are both real algebraic. Our algorithms employ a strengthened form of the NC polynomial-consistency algorithm of Ben-Or, Kozen, and Reif [BKR 1986].

1 Introduction

The subfield $\mathbf{A} \subset \mathbf{R}$ of *real algebraic numbers* consists of real roots α of rational polynomials $p \in \mathbf{Q}[x]$. Real algebraic numbers appear as coordinates of points in problems of computational geometry and robotics. If the algorithms in these areas are to be robust, the underlying computation must be capable of performing exact arithmetic and comparison operations with algebraic numbers. Fixed, finite precision arithmetic may lead to incorrect, or worse, topologically inconsistent answers. We are looking both for efficient representations and fast parallel algorithms (which for us means of class NC) over the subfield $\mathbf{A}$.

The standard representation for $\alpha \in \mathbf{A}$ has been $\alpha \rightarrow (p(x), [r, s])$, where $p \in \mathbf{Q}[x]$ has α as a root, and $[r, s]$ is an *isolating interval* with rational endpoints which uniquely identifies the root α. Thus the isolating interval must be small enough to separate α from all the other roots of $p(x)$. The bit complexity of this representation depends on the heights of the coefficients of $p(x) = \sum a_i x^i$, and the number of bits needed to specify the isolating interval.

The best known bound on the root separation as a function of coefficient size is given by

$$\mathrm{Sep}(p) > \sqrt{3}\, d^{-(d+2)/2} ||p||^{1-d}, \qquad (1)$$

where $d = \deg(p)$, and $||p|| = (\sum a_i^2)^{1/2}$. (See [Mig 1982], who also points out the example of an irreducible polynomial $p(x) = x^d - 2(ax-1)^2$ which has $\mathrm{Sep}(p) < 2a^{-(d+2)/2}$.) Lower bound (1) implies that the representation based on isolating intervals can be presented with $O(d(\log d + \log ||p||))$ many bits for the isolating interval. However, the process of finding an isolating interval using bisection (or multisection) does not seem to parallelize well, hence an alternative representation is sought.

We propose to use the *sign-representation* of real algebraic numbers. We describe explicit poly-log depth circuits for doing exact arithmetic and sign determination.

2 Sign-representation of real algebraic numbers

It is convenient to view the sign of a real number $\mathrm{sgn}(x) \in \{-1, 0, +1\}$ as an element of $\mathbf{Z}/3\mathbf{Z}$. We speak of "sign-vectors" $\mathbf{s} \in (\mathbf{Z}/3\mathbf{Z})^n$ and "sign-arrays" $\mathbf{s} \in Mat_{m,n}(\mathbf{Z}/3\mathbf{Z})$ (the m by n matrices with coefficients in $\mathbf{Z}/3\mathbf{Z}$).

Definition 2.1 *Given* $\mathbf{q} = [q_1, \ldots, q_n] \in \mathrm{Q}[x]^n$, *and a vector* $\mathbf{r} = [r_1, \ldots, r_m]$ *of real roots of*

$p \in \mathbf{Q}[x]$, *define the sign-array* $\mathrm{s}_p(\mathbf{r}, \mathbf{q}) \stackrel{\text{def}}{=} (\mathrm{sgn}(q_i(r_j))) \in Mat_{m,n}(\mathbf{Z}/3\mathbf{Z})$. *If there is only one root* α *in question, then we get a sign-vector* $\mathrm{s}_p(\alpha, \mathbf{q})$. *If the underlying polynomial* $p(x)$ *is understood, then the subscript may be suppressed.*

Definition 2.2 *The "sign-representation" of the real algebraic root* α *of the of the polynomial* $p \in \mathbf{Q}[x]$, *of degree* n, *is*

$$\langle \alpha \rangle \stackrel{\text{def}}{=} (p, [sgn(p'(\alpha)), \ldots, sgn(p^{(n-1)}(\alpha))]). \quad (2)$$

The second component identifies α *among the roots of* $p(x)$ *as we shall see. Rational numbers* $r \in \mathbf{Q}$ *are represented as* $(x - r, \emptyset)$. *We denote the vector* $[p(x), \ldots, p^{(n-1)}(x)] \in \mathbf{Q}[x]^{n-1}$ *by* $\mathcal{F}[p]$, *the "Fourier sequence" (in our case we are using* $\mathcal{F}[p']$*).*

This representation appears in the work of Coste and Roy [CoR 1988]. They point out that *Thom's lemma* may be applied to univariate polynomials to guarantee the soundness of the representation. The following definition is collected here for convenience:

Definition 2.3 *Given* $\mathbf{q} = [q_1, \ldots, q_n] \in \mathbf{Q}[x]^n$, $p \in \mathbf{Q}[x]$, *and* $\mathbf{s} \in (\mathbf{Z}/3\mathbf{Z})^n$ *a sign-vector, define the "sign-condition"*

$$C_p(\mathbf{q} = \mathbf{s}) \stackrel{\text{def}}{=} \{x \in \mathbf{R} \mid p(x) = 0 \wedge \mathrm{sgn}(q_i(x)) = \mathrm{s}[i],\ i = 1, \ldots, n\}. \quad (3)$$

That is to say, $C_p(\mathbf{q} = \mathbf{s})$ *is the subset of roots of* $p(x)$ *where the vector* $\mathbf{q}$ *adopts the sign pattern* $\mathbf{s}$. *In case* $p(x)$ *is understood, the subscript may be suppressed.*

Note. We apply the convention that lower case symbols $c_p(\mathbf{q} = \mathbf{s})$, $c(q = [+1])$, etc.., refer to cardinalities of the sets denoted by their corresponding upper case versions. A useful mnemonic is that subscripted $\mathbf{s}$ symbols refer to sign patterns, whereas subscripted C and c symbols refer to *C*onditions which determine subsets of roots.

3 Soundness of the sign representation

Theorem 3.1 *Given* α, α' *two real roots of the polynomial* $p(x) \in \mathbf{R}[x]$, *then* $\langle \alpha \rangle = \langle \alpha' \rangle$ *implies* $\alpha = \alpha'$.

Proof: It is sufficient to show that $\mathrm{s}_p(\alpha, \mathcal{F}[p']) = \mathrm{s}_p(\alpha', \mathcal{F}[p']) \Rightarrow \alpha = \alpha'$, which follows directly from the following lemma and its corollary [CoR 1988].

Lemma 3.2 *(Little Thom's lemma) Suppose* $p \in \mathbf{R}[x]$, $deg(p) = n > 0$, *and* $\mathbf{s}$ *is a sign-vector of length* n, *indexed from* 0 *to* $n - 1$. *Let*

$$A(\mathbf{s}) = \{x \in \mathbf{R} \mid \mathrm{sgn}(p^{(i)}(x)) = \mathrm{s}[i],\ i = 0, \ldots, n-1\}.$$

Then $A(s) = \emptyset$ *or* $A(s)$ *is connected. (In fact, the sign conditions may include* ≤ 0 *or* ≥ 0, *but we shall not need these here.)*

Corollary 3.3 *Suppose* $p(x) \in \mathbf{R}[x]$ *is of degree* $n > 0$, *and let* α, α' *be two real roots of* $p(x)$. *Suppose* $0 \leq m < n$. *If* $p^{(m)}(\alpha) = p^{(m)}(\alpha') = 0$ *and* $\mathrm{sgn}(p^{(m+k)}(\alpha)) = \mathrm{sgn}(p^{(m+k)}(\alpha'))$ *for* $1 \leq k \leq n - m - 1$, *then* $\alpha = \alpha'$

The proof of theorem 3.1 now follows, since for any two roots α and α' of $p(x)$, at least $p^{(0)}(\alpha) = p^{(0)}(\alpha')$, and the signs of the rest of the derivatives are equal by assumption. □

The sign-sequence representation cannot easily be made canonical, since that would require finding in all cases a unique representing polynomial for each algebraic number, which would necessarily be its irreducible polynomial. We bypass this problem by providing efficient parallel algorithms for doing equality comparisons.

The basic tool which permits the rapid determination of $\langle \alpha \rangle$ is the NC algorithm of Ben-Or, Kozen, and Reif [BKR 1986].

Algorithm $BKR(p, [q_1, \ldots, q_n])$

Input: $p \in \mathbf{Q}[x]$, $\mathbf{q} = [q_1, \ldots, q_n] \in \mathbf{Q}[x]^n$.

Output: $B = \{\mathbf{s} \in (\mathbf{Z}/3\mathbf{Z})^n \mid C_p(\mathbf{q} = \mathbf{s}) \neq \emptyset\}$, $C = \{c_p(\mathbf{q} = \mathbf{s}) \mid \mathbf{s} \in B\}$.

That is to say, all the sign-vectors $\mathbf{s}$ such that $C_p(\mathbf{q} = \mathbf{s}) \neq \emptyset$, and the corresponding cardinalities.

4 Sign-counting Lemma

The BKR algorithm depends upon an application of Sturm's theorem, which we wish to present in a fully generalized form requiring no special assumptions regarding the input polynomials.

Definition 4.1 *For $p, q \in \mathbf{Q}[x]$, define the "generalized Sturm sequence"* $\text{Sturm}(p,q) = [r_0 = p, r_1 = q, r_2, \ldots, r_n]$ *as the vector of successive negative remainders which begins with p and q, (i.e.) $r_{j-1}(x) = q_j(x)r_j(x) - r_{j+1}(x)$ and $r_{n-1}(x) = q_n(x)r_n(x)$.*

Definition 4.2 *Define the "sign variation" of a sign-vector $\mathbf{s} \in (\mathbf{Z}/3\mathbf{Z})^n$ to be the number of times the components change sign when read from either end in sequential order, ignoring zeros. Denote this by* $\text{Var}[\mathbf{s}]$. *For a vector $\mathbf{q} \in Q[x]^n$ and $a \in \mathbf{R}$, let* $\text{Var}_a[\mathbf{q}]$ *be* $\text{Var}[\mathbf{q}(a)]$.

Lemma 4.3 *Suppose the sign vector $\mathbf{s}$ contains no zeros and is covered by subintervals $C_i = [a_i, b_i]$. $\mathbf{s} = [s_1, \ldots, s_n] = C_1 \cup C_2 \cup \ldots \cup C_k$, where $(\forall i)\ C_{i-1} \cap C_i = \{b_{i-1}\} = \{a_i\}$. Then $\sum_{i=1}^{k} \text{Var}[C_i] = \text{Var}[\mathbf{s}]$.*

Definition 4.4 *For $p, q \in \mathbf{Q}[x]$, define the "Sturm query" $S(p,q)$ to be*

$$\text{Var}_{-\infty}[\text{Sturm}(p,q)] - \text{Var}_{+\infty}[\text{Sturm}(p,q)].$$

Lemma 4.5 *Given $p \in \mathbf{Q}[x]$, not necessarily square-free, then the Sturm query $S(p,p')$ computes the total number of real roots of $p(x)$ ignoring multiplicity.*

Proof: Omitted. □

Note that when sweeping past a root of r_n all the terms r_j will become simultaneously zero, but we have shown that the sign pattern of the sequence $[r_0, \ldots, r_n]$ must emerge sorted out with exactly one loss of sign variation between $p(x)$ and $p'(x)$. (The $r_i(x)$ are Sturm remainder polynomials.)

Lemma 4.6 *With the above notations and those of definition (2.3), we have for $p, q \in \mathbf{Q}[x]$*

$$S(p, p'q) = c_p(q = [+1]) - c_p(q = [-1]) \qquad (4)$$

Proof: Omitted. □

5 *BKR* Identity

The second basic ingredient needed for the *BKR* algorithm is the matrix identity of Ben-Or, Kozen, and Reif, for which we present a significantly simplied proof. In the following discussion R and S denote commutative rings.

Definition 5.1 *For $A \in Mat_{m_1,n_1}(S)$ and $B \in Mat_{m_2,n_2}(S)$, define the "Kronecker product" $A \times B \in Mat_{m_1m_2,n_1n_2}(S)$ as the matrix consisting of $m_1 \times n_1$ blocks each of size $m_2 \times n_2$, gotten by replacing each component a_{ij} of A by $a_{ij}B$. We may apply the Kronecker product to vectors as a special case.*

Lemma 5.2 *Suppose $A \in Mat_n(S)$ and $B \in Mat_m(S)$, $v \in S^n$ and $w \in S^m$, then $(A \times B)(v \times w) = (Av) \times (Bw)$.*

Fix $p \in \mathbf{Q}[x]$, with real roots $\{r_1, \ldots, r_k\}$.

In the following discussion let S be the real algebra $\mathbf{R}^k$ (under componentwise multiplication) in which the bit vectors $\mathbf{Z}/2\mathbf{Z}^k$ are embedded. These may be used to represent subsets of roots of $p(x)$ specified by sign-conditions. We observe that the componentwise product represents intersection of subsets.

Let R be the algebra $\mathbf{R}^k$ (again), but considered to contain $\mathbf{Z}/3\mathbf{Z}^k$, and used to represent sign-vectors $\mathbf{s}_p(r_j, q)$. We observe that the componentwise product of two sign-vectors $\mathbf{s}, \mathbf{s}'$ corresponding to $q, q' \in \mathbf{Q}[x]$ represents the sign-vector of the product qq'.

Lemma 5.3 *Suppose $c \in S^n$ and $c' \in S^m$ are vectors of bit vectors specifying n and m subsets of roots of $p \in \mathbf{Q}[x]$, respectively. Suppose $\mathbf{s} \in R^n$ and $\mathbf{s}' \in R^m$ are vectors of sign-vectors specifying sign patterns achieved at the roots of p by n and m polynomials of the form $\prod_{i \in I_\alpha} q_i$ and $\prod_{j \in J_\beta} q_j$, respectively, for n subsets $\{I_\alpha\}$ and m subsets $\{J_\beta\}$. Then for any $A \in Mat_n(\mathbf{R}), A' \in Mat_m(\mathbf{R})$*

$$Ac = s,\ A'c' = s' \ \Rightarrow\ (A \times A')(c \times c') = s'',$$

where $s'' \in R^{nm}$ is a vector whose components are the mn sign-vectors of $(\prod_{i \in I_\alpha} q_i)(\prod_{j \in J_\beta} q_j)$.

Proof: This follows directly from lemma 5.2 and the observations above. □

The ultimate object is to determine which sign conditions on the polynomials $\{q_j\}$ are consistent, (i.e.) represent non-empty sets of roots of $p(x)$. Consistency can easily be checked by applying the linear functional $\mu([x_1, \ldots, x_k]) = \sum_{j=1}^{k} x_j$ to both sides of the linear relations, and using the fact that $\mu^{(n)}(Ac) = A\mu^{(n)}(c)$. Now, μ applied to a bit vector in G counts the size of the set, whereas

μ applied to a sign sequence in H computes the difference between the number of $+1$'s and the number of -1's, ignoring zeros. The sign-counting lemma 4.6 says we may evaluate μ of a sign-vector using Sturm queries: $\mu(\mathbf{s}_p(q)) = S(p, p'q)$, whereas the terms $\mu(c \times c')$ on the left hand side correspond directly to the cardinalities of the subsets of roots determined by conjunctions of sign conditions. By combining the sign conditions and sign vectors pairwise in a log-depth tree we eventually test every possible sign combination for consistency.

6 The BKR circuit

The BKR function is computed by a log-depth circuit. At the leaves one determines all the values $c(q_j = [0])$, $c(q_j = [+1])$, and $c(q_j = [-1])$ by applying the linear functional μ to both sides of the linear relation of the following lemma:

Lemma 6.1 *Given $p, q \in \mathbf{Q}[x]$, then*

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 1 & -1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} C_p(q = [0]) \\ C_p(q = [-1] \\ C_p(q = [+1]) \end{bmatrix} = \begin{bmatrix} \mathbf{1} \\ \mathbf{s}_p(q) \\ \mathbf{s}_p(q^2) \end{bmatrix},$$

which is a linear relation between bit vectors on the left hand side and sign vectors on the right.

Proof: The first row just says that the union of the subsets of roots of p where q is zero, positive, and negative respectively, accounts for all the roots of p. Note that $\mu(\mathbf{1}) = S(p, p')$, the ordinary Sturm query. The second row just restates the definition of $\mathbf{s}_p(q)$. The third row says that $C_p(q = [+1]) + C_p(q = [-1])$ equals the vector of signs of q^2 at the roots of p, which is correct because q^2 is positive when q is either positive or negative. □

The remaining stages of the BKR circuit involve composing pairs of linear relations $Ac = s$ and $Ac' = s'$ using the BKR matrix identity to get new relations $(A \times A')(c \times c') = s''$ as in lemma 5.3. After each stage one applies μ to both sides, which reduces on the right to some more Sturm queries via the sign-counting lemma.

After solving $\mu(c \times c') = (A \times A')^{-1})\mu(s'') = (A^{-1} \times A'^{-1})\mu(s'')$, one may reduce the original system to rank at most k (since the number of roots of $p(x)$ bounds the number of consistent sign-conditions). The result is a new system of the form $A''c'' = s'''$ which is ready for another application of the basic step.

7 Generalized BKR circuit

In order to describe algorithms for the field operations, we need a generalization of BKR to pairs of polynomials in two variables.

Circuit
$BKR2(p_1(x),\ p_2(y),\ [q_1(x,y), \ldots, q_n(x,y)]$

Input: $p_1(x) \in \mathbf{Q}[x]$, $p_2(y) \in \mathbf{Q}[y]$, $\mathbf{q} = [q_1(x,y), \ldots, q_n(x,y)] \subset \mathbf{Q}[x,y]^n$.

Output: Sign sequences of $\mathbf{q}$ at all solutions (α, β) of p_1, p_2, with multiplicities.

7.1 Specification of $BKR2$

View the q_j as polynomials in y, $q_j(x,y) = q_{j,d_j}(x)y^{d_j} + \ldots + q_{j,0}(x), j = 1, \ldots, n$.

1. It is possible that certain leading coefficients $q_{j,d_j}(x)$ may vanish at a root α of $p_1(x)$. It is necessary to determine when this happens because it alters the effective degree of the polynomial $q_j(\alpha, y)$, hence the form of the Sturm query to be performed. So compute

$$\begin{aligned} &BRK(p_1(x), [q_{1,1}(x), \ldots, q_{1,d_1}(x)]), \\ &BRK(p_1(x), [q_{2,1}(x), \ldots, q_{2,d_2}(x)]), \\ &\qquad\vdots \\ &BRK(p_1(x), [q_{n,1}(x), \ldots, q_{n,d_n}(x)]). \end{aligned}$$

   This tells us which polynomials are zero at roots of $p_1(x)$ (among other things), hence which "reductum" of each $q_j(x,y)$ to use at each different root of $p_1(x)$. In the worst case a different reductum is required for each of $O(N)$ roots of $p_1(x)$, so we may have increased the number of polynomials by a factor $O(N)$. Relabel the q_j to refer to this expanded set of polynomials.

2. For each j compute the integral Sturm sequences

$$\begin{aligned} Sturm(p_2, p_2') &= [u_0, u_1, \ldots, u_\delta], \\ Sturm(p_2, p_2'q_j) &= [v_0, v_1, \ldots, v_\epsilon], \\ Sturm(p_2, p_2'q_j^2) &= [w_0, w_1, \ldots, w_\eta], \end{aligned}$$

   in $\mathbf{Q}[x][y]$. This may be done using principal minors of the Sylvester resultant, which we describe subsequently. Note that the cost

of multiplying matrix components is now logarithmic, since they may be polynomials in $\mathbf{Q}[x]$. The same values arise whether one sets $x = r$ then computes the determinants, or first computes the polynomial determinants, and then sets $x = r$. Therefore to compute these Sturm queries, it is sufficient to know the signs of the leading coefficients of u_i, v_j, and w_k at the roots of $p_1(x)$.

3. Compute

$$BKR(p_1, [\{\text{Hcoeff}(u_i)\}, \{\text{Hcoeff}(v_j)\}, \{\text{Hcoeff}(w_k)\}])$$

where Hcoeff denotes the head coefficient of a polynomial. This tell us, for each root $p_1(\alpha) = 0$, which sign sequences are realized by the collection of head coefficients appearing in the Sturm queries above.

4. We now have sufficient information to compute $c_{p_2}(q_j = [0])$, $c_{p_2}(q_j = [+1]$, and $c_{p_2}(q_j = [-1])$ for each j.

5. Combine these results using the BKR matrix identity exactly as before. New Sturm queries $S(p_2, p_2'(\prod_{i\in I} q_i)(\prod_{j\in J} q_j))$ will appear on the right hand side as one goes along. These must be evaluated by additional recursive calls to BKR as above to determine the proper reducta and Sturm coefficients.

6. Continue through logarithmically many recursive stages until the full answer is computed.

The two variable generalization has the basic form of a $\log^2$-depth tree, where there are numerous subcircuits to perform polynomial multiplication, Sturm queries via determinants, linear equation solving, and basis extraction.

8 Circuits for real algebraic arithmetic

8.1 Sign-vectors for the roots of f

Call $BKR(f,\ \mathcal{F}[f'])$; Thom's lemma assures us that any sign-vector which is realized will be realized only once.

8.2 Sign of g at one root of f

Call $BKR(f,\ [\mathcal{F}[f'],\ g])$; the rationale is that among the sign-vectors realized by $[\mathcal{F}[f'],\ g]$ at the roots of f, the sign-vector of α will appear uniquely among the initial segments $\mathcal{F}[f']$. So run BKR, then do a prefix match.

8.3 Sign-representation for $-\alpha$

If $\langle\alpha\rangle = (f(x), [s_1, s_2, \ldots, s_{n-1}])$ then $\langle-\alpha\rangle = (f(-x), [s_1, -s_2, \ldots, (-1)^n s_{n-1}])$.

8.4 Sign-representation for $\alpha + \beta$

Recall that $r(x) = Res_y(f(x-y), g(y))$, the resultant with respect to y, has all the sums $\alpha_i + \beta_j$ as roots, where $\{\alpha_i\}$ are the roots of f, and $\{\beta_j\}$ are the roots of g. Next, compute

$$BKR2(f(x),\ g(y),\ [\mathcal{F}[f'](x),\ \mathcal{F}[g'](y),\ \mathcal{F}[r'](x+y)]).$$

This gives us the list of sign-vectors achieved by $[\mathcal{F}[f'](x),\ \mathcal{F}[g'](y),\ \mathcal{F}[r'](x+y)]$ at the solutions (α_i, β_j) of the system $\{f(x), g(y)\}$. We may identify the mapping $(\langle\alpha_i\rangle, \langle\beta_j\rangle) \mapsto \langle\alpha_i + \beta_j\rangle$ by doing a prefix search for $\mathrm{s}_f(\alpha_i, \mathcal{F}[f'])$ and $\mathrm{s}_g(\beta_j, \mathcal{F}[g'])$.

8.5 Sign-representation for $\alpha\beta$

Recall that the polynomial $r(x) = Res_y(y^m f(x/y), g(y))$ has all the products $\alpha_i\beta_j$ as roots. Compute

$$BKR2(f(x),\ g(y),\ [\mathcal{F}[f'](x),\ \mathcal{F}[g'](y),\ \mathcal{F}[r'](xy)]).$$

This gives us the sign-vectors achieved by $[\mathcal{F}[f'](x),\ \mathcal{F}[g'](y),\ \mathcal{F}[r'](xy)]$ at the solutions (α, β) of the system $\{f(x), g(y)\}$. Again, we can identify the mapping $(\langle\alpha_i\rangle, \langle\beta_j\rangle) \mapsto \langle\alpha_i\beta_j\rangle$ by doing a prefix search for $\mathrm{s}_f(\alpha_i, \mathcal{F}[f'])$ and $\mathrm{s}_g(\beta_j, \mathcal{F}[g'])$.

8.6 $\alpha <, =, > \beta$

Compute

$$BKR2(f(x),\ g(y),\ [\mathcal{F}[f'](x),\ \mathcal{F}[g'](y),\ (x-y)]).$$

For each solution (α, β) of $\{f(x), g(y)\}$ this prefixes the sign of $x - y$ with $\mathrm{s}_f(\alpha_i, \mathcal{F}[f'])$ and $\mathrm{s}_g(\beta_j, \mathcal{F}[g'])$, which is sufficient to decide $x > y$, $x = y$, or $x < y$.

8.7 Sign-representation for $1/\alpha$

First determine the sign of α by applying the algorithm which determines the sign of $g(x) = x$ at the roots of f. If $\alpha = 0$, then stop. Otherwise, the reversed polynomial $r(x) = x^m f(1/x)$ has the inverses $1/\alpha_i$ as roots.

Consider $\mathcal{F}[f'](1/x)$; we would like to multiply each component by a sufficiently high *even* power of x so that it returns to being a polynomial without changing its sign. Call the resulting sequence $\sigma(x)$. Next compute

$$BKR(r(x), [\sigma(x), \mathcal{F}[r']]).$$

This prefixes the sign codes at the roots $1/\alpha$ of r with the signs acquired by $\mathcal{F}[f']$ at α.

8.8 Sign-representation for $r + \alpha$ and $r \cdot \alpha$

Here $r \in \mathbf{Q}$. The polynomial $f(x - r)$ has $r + \alpha$ as a root, and the signs of its derivatives at $\alpha + r$ are identical to those of $f(x)$ at α.

The polynomial $g(x) = r^m f(x/r)$ has $r \cdot \alpha$ as a root. In this case

$$\mathcal{F}[g'](x) = \\ [r^{m-1} f'(x/r), r^{m-2} f''(x/r), \ldots, r f^{(m-1)}(x/r)],$$

and the signs depend on $\langle \alpha \rangle$, the sign of r, and the parity of the exponents (in case r is negative).

9 Complexity of the circuits

There are $\log N$ parallel stages, where each combining step entails $O(N^2)$ additional Sturm queries on the right hand side. As remarked earlier, the linear systems so generated may be reduced to size $O(N)$. At each stage the system has to be solved, then pared down by eliminating the columns corresponding to zero entries in the c-vector, and finally run through a subcircuit which extracts a square sub-system of the reduced rank.

The $O(N^2)$ additional Sturm queries have sizes bounded by $O(N^2)$ and therefore circuit depth still bounded by $O(\log^2 N)$. The circuit size grows, however, to be $O(N^{C+3})$, where $O(N^C)$ is the best available size for an $O(\log^2 N)$-depth circuit which computes the determinant. Currently $C \leq \alpha + 1 + \epsilon$, where α is the exponent of N for the best available $O(\log N)$-depth circuit which multiplies two $N \times N$ matrices, and $\epsilon > 0$ [Ber 1984].

Both the linear system solution and the paring down with extraction of a square subsystem can be reduced to the rank problem. (One computes in parallel the ranks of the submatrices of rows 1 through k and retains the rows where the rank increases.) K. Mulmuley [Mul 1987] shows how to find the rank of an $n \times m$ matrix A by computing a characteristic polynomial $Q(t) = det(tI - XA)$. Applying the result of Berkowitz we may obtain a circuit of depth $O(\log^2 N)$ and size $O(N^C)$ for the rank problem.

There are $O(\log N)$ stages in the basic BKR circuit, so the overall depth bound is $O(\log^3 N)$, and the size bound is $O(N^{C+3})$.

The depth complexity of our circuit for computing $BKR2$ continues to be dominated by the cost of Sturm queries. These queries involve determinants of matrices with univariate polynomial entries. Their sizes may be estimated using Hadamard's inequality, and bounded as $O(N^2)$. This will not affect the circuit depth, but it will increase the circuit sizes. We must also concern ourselves with the number and degree of the polynomials being presented to the recursive subcircuits. Both the number and degrees of the polynomials may grow quadratically. In both cases, as with the coefficient size, we have recursive inputs which are $O(N^2)$.

The circuit depth for $BKR2$ increases by a factor of $\log N$, since it becomes necessary to put in polynomial multiplication circuits where previously there had been only integer multiplication circuits. Also, each outer Sturm query entails $O(N^2)$ recursive applications of the basic BKR circuit to determine signs of leading coefficient polynomials. These additional factors multiply together to produce an overall depth bound of $O(\log^7 N)$ for the BKR circuit. The size increases to $O(N^2 N^{C+3}) = O(N^{C+5})$. Our experience with practical implementation indicates that the computation parallelizes as expected, and can be effective in situations in which the degrees of the algebraic numbers involved in the computation can be bounded in advance.

10 Summary of circuit complexity for real algebraic arithmetic

In all cases where either BKR or $BKR2$ are called, they dominate the complexity. We understand α and β to be roots of $f \in \mathbf{Q}[x]$, $r \in \mathbf{Q}$ to be a rational constant, and $O(N^C)$ to be the best available size for an $O(\log^2 N)$-depth circuit which computes the determinant.

Summary		
Stage	**depth**	**size**
Outer product	$O(1)$	$O(N^4)$
Sturm queries	$O(\log^2 N)$	$O(N^{C+3})$
System solution	$O(\log^2 N)$	$O(N^{C+1})$
Reduction	$O(\log^2 N)$	$O(N^{C+1})$
BKR	$O(\log^3 N)$	$O(N^{C+3})$
$BKR2$	$O(\log^7 N)$	$O(N^{C+5})$
Operation	**depth**	**size**
Roots of f	$O(\log^3 N)$	$O(N^{C+3})$
$\mathrm{sgn}(g(\alpha))$	$O(\log^3 N)$	$O(N^{C+3})$
$\langle -\alpha \rangle$	$O(1)$	$O(N)$
$\langle \alpha + \beta \rangle$	$O(\log^7 N)$	$O(N^{C+5})$
$\langle \alpha \cdot \beta \rangle$	$O(\log^7 N)$	$O(N^{C+5})$
$\alpha <, =, > \beta$	$O(\log^7 N)$	$O(N^{C+5})$
$\langle 1/\alpha \rangle$	$O(\log^3 N)$	$O(N^{C+3})$
$\langle r + \alpha \rangle$	$O(1)$	$O(N)$
$\langle r \cdot \alpha \rangle$	$O(1)$	$O(N^2)$

11 References

[Akr 1980] Akritas, A. G.: "Elements of Computer Algebra, With Applications", Wiley Interscience, 1989.

[Bar 1968] Bareiss, E. H.: "Sylvester's Identity and Multistep Integer-preserving Gaussian Elimination", Math. Comp. **22** (1968), 565-578.

[BKR 1986] Ben-Or, M., D. Kozen, and J. Reif: "The Complexity of Elemntary Algebra and Geometry", Journal of Computer and System Sciences **32** (1986), 251-264.

[Ber 1984] Berkowitz, M.: "On computing the determinant in small parallel time using a small number of processors", Information Processing Letters **18** (1984), 147-150.

[BCR 1987] Bochnak, J., M. Coste, M-F. Roy: "Géométrie algebrique réelle", A series of Modern Surveys in Mathematics, Vol. 12 (1987), Springer Verlag, Berlin.

[Chi 1985] Chistov, A. L.: "Fast parallel calculation of the rank of matrices over a field of arbitrary characteristic", in "Proc. Int. Conf. Foundations of Computation Theory", Lecture Notes in Computer Science **199** (1985), 63-69, Springer Verlag.

[CoR 1988] Coste, M., M-F. Roy: "Thom's lemma, the coding of real algebraic numbers, and the computation of the topology of semi-algebraic sets", J. Symbolic Computation (1988) **5**, 121-129.

[Csa 1976] Csanky, L.: "Fast parallel matrix inversion algorithms", SIAM J. Comput. Vol. 5, No. 4, December 1976.

[CLR 1989] Cucker, Felipe, Hervé Lanneau, M-F. Roy: "NC Computations with Real Algebraic Numbers", SIAM J. Comp. (submitted).

[Gat 1984] von zur Gathen, J.: "Parallel algorithms for algebraic problems", SIAM J. Comput. Vol. 13, No. 4, November 1984.

[Loo 1982a] Loos, R.: "Computing in Algebraic Extensions", in "Computer Algebra, Symbolic and Algebraic Computation", edited by B. Buchberger, G. E. Collins, and R. Loos in cooperation with R. Albrecht, Springer Verlag, Wein, 1982.

[Mig 1982] Mignotte, M.: "Some Useful Bounds", in "Computer Algebra, Symbolic and Algebraic Computation", edited by B. Buchberger, G. E. Collins, and R. Loos in cooperation with R. Albrecht, Springer Verlag, Wein, 1982.

[Mul 1987] Mulmuley, K.: "A fast parallel algorithm to compute the rank of a matrix over an arbitrary field", Combinatorica **7** (1) (1987), 101-104.

[Rei 1983] Reif, J.: "Logarithmic depth circuits for algebraic functions", in "Proc. 24th Ann. Symposium on Foundations of Comp. Sci.", Tucson (1983), 138-145.

[RS 1988] Roy, M-F., A. Szpirglas: "Complexity of computations with real algebraic numbers", Journal of Symbolic Comp. (to appear).

[Stu 1904] Sturm, C.: "Über die Auflösung der Numerischen Gleichungen (1835)", in the series "Ostwald's Klassiker der Exacten Wissenschaften", Nr. 143, Wilhelm Engelmann, Leipzig (1904).

[Tar 1951] Tarski, A.: "A Decision Method for Elementary Algebra and Geometry", University of California Press, Berkeley (1951).