A Generalized and Improved Constructive Separation Bound for Real Algebraic Expressions

Kurt Mehlhorn and Stefan Schirra

MPI–I–2000–004 November 2000

Author's Address

Kurt Mehlhorn
Max-Planck-Institut für Informatik
Stuhlsatzenhausweg
66123 Saarbrücken
mehlhorn@mpi-sb.mpg.de

Stefan Schirra
Think & Solve Beratungsgesellschaft
Bahnhofstraße 36
66111 Saarbrücken
stschirr@mpi-sb.mpg.de
ssh@think-solve.com

Acknowledgements

Partially supported by ESPRIT project 28155 (GALIA).
Work done while the second author was at Max-Planck-Institut für Informatik.

Abstract

We prove a separation bound for a large class of algebraic expressions specified by expression dags. The bound applies to expressions whose leaves are integers and whose internal nodes are additions, subtractions, multiplications, divisions, k-th root operations for integral k, and taking roots of polynomials whose coefficients are given by the values of subexpressions. The (logarithm of the) new bound depends linearly on the algebraic degree of the expression. Previous bounds applied to a smaller class of expressions and did not guarantee linear dependency.

Keywords

Separation Bound, Exact Computation

1 Introduction

The evaluation of conditions in a computer program frequently amounts to determining the signs of arithmetic expressions. The evaluation of arithmetic expressions with floating point arithmetic (more generally, any imprecise arithmetic) incurs round-off error and hence cannot guarantee the computation of the correct sign. As a consequence, a computation may follow an incorrect path. This may lead to catastrophic errors.

In the realm of geometric computations, the situation is particularly severe and known as the precision caused robustness problem [15, 18, 27, 36, 39]. A popular approach to overcoming the robustness problem is the exact computation paradigm [1, 2, 4, 5, 9, 10, 7, 13, 25, 33, 40, 41]. The paradigm calls for the exact evaluations of all conditions and hence the exact computation of signs.

In this paper, we consider the sign computation for the following class of *real algebraic expressions*. The value of a real algebraic expression is either a real number or undefined (in Section 6 we show how to test whether the value of an expression is defined).

(1) Any integer v is a real algebraic expression. The integer is also the value of the expression.

(2) If E_1 and E_2 are real algebraic expressions, so are E_1+E_2, E_1-E_2, $E_1\cdot E_2$, E_1/E_2, and $\sqrt[k]{E_1}$, where $k \geq 2$ is an integer. The value of $\sqrt[k]{E_1}$ is undefined if k is even and the value of E_1 is negative. The value of E_1/E_2 is undefined, if the value of E_2 is zero. The value of E_1+E_2, E_1-E_2, $E_1\cdot E_2$, E_1/E_2, or $\sqrt[k]{E_1}$ is undefined, if the value of E_1 or the value of E_2 is undefined. Otherwise the value of E_1+E_2, E_1-E_2, $E_1\cdot E_2$, and E_1/E_2 is the sum, the difference, the product and the quotient of the values of E_1 and E_2 respectively and the value of $\sqrt[k]{E_1}$ is the k-th root of the value of E_1.

(3) If E_d, E_{d-1}, $\ldots$, E_1, E_0 are real algebraic expressions and j is a positive integer, then $\diamond(j,E_d,E_{d-1},\ldots,E_1,E_0)$ is an expression. If the values of the E_i are defined, the value of the expression is the j-th smallest real root of the polynomial $\xi_d X^d + \xi_{d-1} + \ldots \xi_0$, if the polynomial has at least j real roots, where ξ_i is the value of E_i. Otherwise, the value is undefined.

Below, expression always means real algebraic expression. An expression is given as a directed acyclic graph (dag) whose source nodes are labeled by the operands and whose internal nodes are labeled by operators.

The class of real algebraic expressions can express precisely the class of real algebraic numbers. Every real algebraic number can be represented by an expression since every real algebraic number is the root of a polynomial with integer coefficients. We simply use this polynomial in an expression of type (3). On the other hand, the real algebraic numbers are closed under the operations arising in (1) - (3).

A *separation bound* $sep(E)$ for an expression E having value ξ is a positive real number with the property that $\xi \neq 0$ implies $|\xi| \geq sep(E)$. Separation bounds allow one to determine the sign of an expression by numerical computation. An approximation $\bar{\xi}$ of ξ and an error bound Δ_{error} with $|\xi - \bar{\xi}| \leq \Delta_{\text{error}}$ is computed. If the absolute value of $\bar{\xi}$ is larger than Δ_{error}, the sign of ξ

is equal to the sign of $\bar{\xi}$. If not, the approximation is computed with higher precision, say as to guarantee halving the error bound. This is repeated until either $|\bar{\xi}| > \Delta_{\text{error}}$ (and hence the sign of ξ is known by the above) or until $|\bar{\xi}| + \Delta_{\text{error}} < sep(E)$, in which case we may conclude that $\xi = 0$. This strategy is used in the number type `leda_real` [5, 8, 28] and the number type `Expr` of the `CORE` package [21].

The efficiency of the sign computation described in the preceding paragraph crucially depends on the quality of the separation bound. Separation bounds have been studied extensively in computer algebra [12, 19, 30, 35, 31, 38], as well as in computational geometry [7, 4, 6, 26, 40, 24, 29]. We review some of this work below.

The starting point for the present work is the bound given by Burnikel et al. [6] for expressions defined by items (1) and (2). We refer to this bound as the BFMS bound in the sequel.

Lemma 1 ([6]) *Let E be an expression with integral operands and operations $+, -, *, /, \sqrt[k]{\ }$ for integral $k \geq 2$. Let ξ be the value of E, let $D(E)$ be the product of the indices of the radical operations in E, and let $u(E)$ and $l(E)$ be defined inductively on the structure of E by the rules shown in the table below.*

	$u(E)$	$l(E)$
integer N	$\lvert N\rvert$	1
$E_1 \pm E_2$	$u(E_1) \cdot l(E_2) + l(E_1) \cdot u(E_2)$	$l(E_1) \cdot l(E_2)$
$E_1 \cdot E_2$	$u(E_1) \cdot u(E_2)$	$l(E_1) \cdot l(E_2)$
E_1 / E_2	$u(E_1) \cdot l(E_2)$	$l(E_1) \cdot u(E_2)$
$\sqrt[k]{E_1}$	$\sqrt[k]{u(E_1)}$	$\sqrt[k]{l(E_1)}$

Then either $\xi = 0$ or

$$\left(l(E)u(E)^{D(E)^2-1}\right)^{-1} \leq |\xi| \leq u(E)l(E)^{D(E)^2-1}.$$

If E is division-free, $l(E) = 1$, and the above bound holds with $D(E)^2$ replaced by $D(E)$.

We give an example to illustrate the dependence on $D = D(E)$. We refer to D as the degree bound of the expression. Observe that D is an upper bound on the algebraic degree of the value of the expression. Consider the expression

$$\frac{2^{10}\sqrt[8]{2^8 - (2^8 - 1)} - 2^6}{1}.$$

Here $u(E) \approx 2^{10}$, $l(E) = 1$ and $D(E) = 8$. So the BFMS bound is $2^{-10\cdot 63} = 2^{-630}$, since E is not division-free and hence the dependence (of the logarithm of the bound) on D is quadratic. Without the final redundant division, the expression is division-free and the bound becomes $2^{-10\cdot 7} = 2^{-70}$. The example shows that linear or quadratic dependency on D makes a tremendous difference for the quality of the separation bound.

Li and Yap [24] and Mehlhorn and Schirra [29] considered a class of expressions more general than the class considered in [6] (though not as general as the one considered in this paper):

they require the expressions E_d to E_0 in (3) to be integral. The bound in [29] is similar to the bound from [6] above; there is one additional rule covering roots of polynomials with integer coefficients. In particular, the bound is linear in D for division-free expressions and depends on D^2, in general. The bound in [24] is identical to the bound in [29] for division-free expressions. For expressions with divisions, the dependence on D is frequently linear but may be worse than quadratic.

In this paper, we prove a bound which is always linear in D. The paper is structured as follows. In Section 2, we review some facts about roots of polynomials and separation bounds. In Section 3, we show that a simple modification of the arguments in [6] gives a bound linear in D for the expressions defined by (1) and (2). In Section 4, we prove our main theorem, the bound for expressions defined by (1), (2), and (3). In Section 5 we compare our bound to other root bounds, in Section 6 we discuss an application to polynomial system solving, and in Section 7 we offer conclusions.

2 Preliminaries

An algebraic integer is the root of a polynomial with integer coefficients and leading coefficient one. The following three Lemmas were already used in [6] and [24].

Lemma 2 *Let α be an algebraic integer and let $\deg(\alpha)$ be the algebraic degree of α. If U is an upper bound on the absolute value of all conjugates of α, then*

$$|\alpha| \geq U^{1-\deg(\alpha)}$$

Proof: The proof is simple. Let d be the degree of α and let $\alpha_1 = \alpha$, α_2, $\dots$, α_d be the conjugates of α. The product of the conjugates is equal to the constant coefficient of the defining polynomial and hence in $\mathbb{Z}$. Thus $|\alpha| \cdot U^{d-1} \geq 1$. ∎

Lemma 3 *Let α and β be algebraic integers Then $\alpha+\beta$, $\alpha\beta$ and $\sqrt[k]{\alpha}$ are algebraic integers.*

Proof: See [17] or [23] or [6, Theorem 4]. ∎

Lemma 4 *Let α and β be algebraic integers and let U_α and U_β be upper bounds on the absolute size of the conjugates of α and β, respectively. Then $U_\alpha + U_\beta$ is an upper bound on the absolute size of the conjugates of $\alpha \pm \beta$, $U_\alpha U_\beta$ is an upper bound on the absolute size of the conjugates of $\alpha\beta$, and $\sqrt[k]{U_\alpha}$ is an upper bound on the absolute size of the conjugates of $\sqrt[k]{\alpha}$.*

Proof: See [6, Lemma 6]. ∎

We also need to cover item (3) in the definition of algebraic expressions.

Lemma 5 *Let* ρ *be the root of a monic polynomial*

$$P(X) = X^n + \alpha_{n-1}X^{n-1} + \alpha_{n-2}X^{n-2} + \cdots + \alpha_0$$

of degree n where the coefficients $\alpha_{n-1}, \alpha_{n-2}, \ldots, \alpha_0$ *are algebraic integers. Then* ρ *is an algebraic integer.*

Proof: This fact is well-known, a proof can, for example, be found in [34, Theorem 2.4]. We include a proof for completeness. The proof uses an argument similar to the proof of Lemma 3. Let $\alpha_j^{(i_j)}$, $1 \le i_j \le \deg(\alpha_j)$, be the conjugates of α_j for $0 \le j \le n-1$ and let $\widehat{\alpha}_j$ be the vector formed by the conjugates of α_j. Consider the polynomial

$$Q(X) = \prod_{i_0}\prod_{i_1}\cdots\prod_{i_{n-1}}(X^n + \alpha_{n-1}^{(i_{n-1})}X^{n-1} + \alpha_{n-2}^{(i_{n-2})}X^{n-2} + \cdots + \alpha_0^{(i_0)}).$$

ρ is a root of $Q(X)$ and $Q(X)$ is symmetric in the $\alpha_j^{(i_j)}$ for all j. The theorem on elementary symmetric function implies that $Q(X)$ is a polynomial in X and the elementary symmetric functions $\sigma_1(\widehat{\alpha}_j)$, ..., $\sigma_{\deg(\alpha_j)}(\widehat{\alpha}_j)$. The elementary symmetric function $\sigma_l(\widehat{\alpha}_j)$ is the coefficient of $X^{\deg(\alpha_j)-l}$ in the minimal polynomial of α_j and hence in $\mathbb{Z}$ (since α_j is an algebraic integer). Thus $Q(X)$ is a monic polynomial in $\mathbb{Z}[X]$ and ρ is an algebraic integer. ∎

We also need bounds for the absolute size of roots of monic polynomials. Such bounds are well known. Let $P(X) = X^n + a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \cdots + a_0$ be a monic polynomial with arbitrary real coefficients, not necessarily integral, and let α be a root of $A(X)$. A *root bound* Ξ is any function of the coefficients of P that bounds the absolute value of α, i.e.,

$$|\alpha| \le \Xi(a_{n-1}, a_{n-2}, \ldots, a_0)$$

We require that Ξ is monotone, i.e., if $|a_i| \le b_i$ for $0 \le i \le n-1$, then

$$\Xi(a_{n-1}, a_{n-2}, \ldots, a_0) \le \Xi(b_{n-1}, b_{n-2}, \ldots, b_0).$$

Examples of root bounds are:

$$\begin{aligned}
|\alpha| &\le 2\max\left(|a_{n-1}|, \sqrt{|a_{n-2}|}, \sqrt[3]{|a_{n-3}|}, \ldots, \sqrt[n]{|a_0|}\right)\\
|\alpha| &\le 1 + \max\left(|a_{n-1}|, |a_{n-2}|, \ldots, |a_0|\right)\\
|\alpha| &\le \max\left(n|a_{n-1}|, \sqrt{n|a_{n-2}|}, \sqrt[3]{n|a_{n-3}|}, \ldots, \sqrt[n]{n|a_0|}\right)\\
|\alpha| &\le \left(\sqrt[n]{2}-1\right)^{-1}\max\left(\frac{|a_{n-1}|}{\binom{n}{1}}, \sqrt{\frac{|a_{n-2}|}{\binom{n}{2}}}, \sqrt[3]{\frac{|a_{n-3}|}{\binom{n}{3}}}, \ldots, \sqrt[n]{\frac{|a_0|}{\binom{n}{n}}}\right)
\end{aligned}$$

A proof of all bounds can be found in [38]. The first bound is called the Lagrange-Zassenhaus bound and the middle two bounds are called the Cauchy bounds.

3 A Modified BFMS Bound

The BFMS bound is quadratic in $D(E)$. We show that a simple modification of the construction in [6] results in a linear dependency in D.

In the proof of the BFMS bound, the expression dag is restructured such that is contains only a single division and this division is the final operation in the dag. More precisely, for every node A in the original dag, there are two nodes A_ν and A_δ in the restructured dag such that the value of A is equal to the quotient of the values of A_ν and A_δ. For the leaves (which stand for integers), the replacement is trivial (we take $A_\nu = A$ and $A_\delta = 1$), for additions and multiplications the rules are obvious, for divisions the role of A_ν and A_δ are interchanged, and in the case of a root-operation, the operation is applied to A_ν as well as A_δ, i.e., each root operation in the original dag gives rise to two root operations in the restructured dag. Therefore $D(E_\nu)$ and $D(E_\delta)$ are at most $D(E)^2$. The bound for division free expressions applies to E_ν and E_δ.

A slight modification in the rule for $\sqrt[k]{\ }$-operations gives a bound which is linear in $D(E)$. We again replace A by a quotient A'_ν/A'_δ, but use a different replacement rule for root operations. Consider $B = \sqrt[k]{A}$. We set $B'_\nu = \sqrt[k]{A'_\nu (A'_\delta)^{k-1}}$ and $B'_\delta = A'_\delta$. Hence $D(E'_\nu)$ and $D(E'_\delta)$ are at most $D(E)$. The rules for computing upper bounds for the values of A'_ν and A'_δ are given below.

Lemma 6 *Let E be an expression with integral operations and operations $+, -, *, /, \sqrt[k]{\ }$ for integral k, let $D(E)$ be the product of the indices of the radical operations in E, and let $u(E)$ and $l(E)$ be defined inductively on the structure of E by the rules shown in the table below.*

	$u'(E)$	$l'(E)$
integer N	$\lvert N\rvert$	1
$E_1 \pm E_2$	$u'(E_1)\cdot l'(E_2) + l'(E_1)\cdot u'(E_2)$	$l'(E_1)\cdot l'(E_2)$
$E_1 \cdot E_2$	$u'(E_1)\cdot u'(E_2)$	$l'(E_1)\cdot l'(E_2)$
E_1/E_2	$u'(E_1)\cdot l'(E_2)$	$l'(E_1)\cdot u'(E_2)$
$\sqrt[k]{E_1}$	$\sqrt[k]{u'(E_1) l'(E_1)^{k-1}}$	$l'(E_1)$

Then either $\xi = 0$ or

$$\left(l'(E)u'(E)^{D(E)-1}\right)^{-1} \le |\xi| \le u'(E)l'(E)^{D(E)-1}$$

Lemma 6 is a special case of our main theorem and hence we do not give a separate proof for it. The separation bound provided by Lemma 6 is frequently much better than the bound in [6] due to the linear dependence on D. It is not always better because $u'(E)$ and $l'(E)$ are larger than the original $u(E)$ and $l(E)$ values of [6]. We give a simple example. Consider the expressions $\sqrt{N_1/N_2}$ and $\sqrt{N_1/N_2} - \sqrt{N_3/N_4}$, where the N_i are positive integers. We have $u(N_1/N_2) = u'(N_1/N_2) = N_1$ and $l(N_1/N_2) = l'(N_1/N_2) = N_2$, $u(\sqrt{N_1/N_2}) = \sqrt{N_1}$, $l(\sqrt{N_1/N_2}) = \sqrt{N_2}$, $u'(\sqrt{N_1/N_2}) = \sqrt{N_1N_2}$, and $l'(\sqrt{N_1/N_2}) = N_2$. The original BFMS-bound for $\sqrt{N_1/N_2}$ is $(\sqrt{N_2}(\sqrt{N_1})^3)^{-1}$ and the modified bound is $(N_2(\sqrt{N_1N_2})^1)^{-1}$, i.e., the bounds are incomparable. For $E = \sqrt{N_1/N_2} - \sqrt{N_3/N_4}$, we have $u(E) = \sqrt{N_1N_4} + \sqrt{N_2N_3}$, $l(E) = \sqrt{N_2N_4}$, $u'(E) =$

$\sqrt{N_1N_2}N_4+\sqrt{N_3N_4}N_2$, and $l'(E)=N_2N_4$. The original BFMS-bound for $\sqrt{N_1/N_2}-\sqrt{N_3/N_4}$ is $(\sqrt{N_2N_4}(\sqrt{N_1N_4}+\sqrt{N_2N_3})^{15})^{-1}$ and the modified bound is $(N_2N_4(\sqrt{N_1N_2}N_4+\sqrt{N_3N_4}N_2)^3)^{-1}$. Assuming that $N_i \approx 2^k$ the original bound is $2^{-16k-15}$ and the modified bound is 2^{-8k-3}.

4 The General Bound

We derive a separation bound for the expressions defined by (1), (2), and (3). For pedagogical reasons we proceed in two steps. We first consider division-free expressions and only monic polynomials in (3), i.e., we restrict E_d to 1 in (3). These restrictions guarantee that the values of our expressions are algebraic integers. In the second step, we treat the general case. In either case the goal is to extend the table given in Lemma 6 by a rule covering (3).

4.1 Algebraic Integers

We start with the algebraic integer case. Let ρ be the root of a monic polynomial

$$P(X)=X^d+\alpha_{d-1}X^{d-1}+\alpha_{d-2}X^{d-2}+\cdots+\alpha_0$$

of degree d where the coefficients $\alpha_{d-1},\alpha_{d-2},\ldots,\alpha_0$ are algebraic integers, given by expressions $E_{d-1},\ldots,E_0$. The expressions are division-free and use the diamond-operator only with leading coefficient one. The size of ρ and all its conjugates is bounded by $\Xi(\alpha_{d-1},\alpha_{d-2},\ldots,\alpha_0)$ which in turn is bounded by $\Xi(u(E_{d-1}),u(E_{d-2}),\ldots,u(E_0))$. We therefore define $u(E)$ as the latter quantity.

Lemma 7 *Let E be an expression with integer operands and operations $+,-,*,\sqrt[k]{\ }$, for integral k and $\diamond(j,1,\ldots)$ operations*[1]. *Let ξ be the value of E and let* $\deg(\xi)$ *denote the algebraic degree of ξ. Let $u(E)$ be defined inductively on the structure of E according to the following rules.*

E	$u(E)$
integer N	$\lvert N\rvert$
$E_1 \pm E_2$	$u(E_1)+u(E_2)$
$E_1 \cdot E_2$	$u(E_1)\cdot u(E_2)$
$\sqrt[k]{E_1}$	$\sqrt[k]{u(E_1)}$
$\diamond(j,1,E_{d-1}\ldots,E_0)$	$\Xi(\ldots,u(E_i),\ldots)$

Then either $\xi=0$ or

$$\left(u(E)^{\deg(\xi)-1}\right)^{-1} \le |\xi| \le u(E).$$

Lemma 7 is a special case of our main theorem and hence we only sketch the proof: One proves by induction on the structure of E that there is a monic polynomial $P_E(X)\in\mathbb{Z}[X]$ such that $P_E(\xi)=0$ and $|\beta|\le u(E)$ for all roots β of P_E.

[1] Observe that the leading coefficient is restricted to be equal to one.

Why do we give a separate rule for the root-operation. After all, it is a special case of the diamond-operation; the k-th root operation applied to an expression E corresponds to the diamond operation applied to the polynomial $X^k - E$. We treat the root operation separately because general root bounds Ξ give weaker bounds than our special rule. The first Cauchy bound gives us only $1 + u(E)$ instead of $\sqrt[k]{u(E)}$. The other bounds are off by a factor of 2, $\sqrt[k]{k}$, and $\left(\sqrt[k]{2} - 1\right)^{-1}$, respectively.

4.2 The General Case

The diamond operation allows one to take the root of a polynomial

$$P(X) = \alpha_d X^d + \alpha_{d-1} X^{d-1} + \cdots + \alpha_1 X + \alpha_0$$

where the α_i are arbitrary real algebraic numbers. Every real algebraic number can be written as the quotient of two algebraic integers; this is well-known, but will be reproved below as part of the proof of of our main theorem. Let $\alpha_i = \nu_i/\delta_i$ where ν_i and δ_i are algebraic integers. Then

$$P(X) = \frac{\nu_d}{\delta_d} X^d + \frac{\nu_{d-1}}{\delta_{d-1}} X^{d-1} + \cdots + \frac{\nu_1}{\delta_1} X + \frac{\nu_0}{\delta_0}$$

Let $D = \prod \delta_i$. By multiplication with D we obtain

$$D \cdot P(X) = (\nu_d D/\delta_d) X^d + (\nu_{d-1} D/\delta_{d-1}) X^{d-1} + \cdots + (\nu_1 D/\delta_1) X + (\nu_0 D/\delta_0),$$

a polynomial whose coefficients are algebraic integers. We next derive a monic polynomial. To get rid off the leading coefficient $(\nu_d D/\delta_d)$, we multiply by $(\nu_d D/\delta_d)^{d-1}$ and substitute $X/(\nu_d D/\delta_d)$ for X. We obtain

$$D \cdot (\nu_d D/\delta_d)^{d-1} \cdot P\left(\frac{X}{\nu_d D/\delta_d}\right) = Q(X) =$$

$$X^d + (\nu_d D/\delta_d)(\nu_{d-1} D/\delta_{d-1}) X^{d-1} + \cdots + (\nu_d D/\delta_d)^{d-1} (\nu_1 D/\delta_1) X + (\nu_d D/\delta_d)^d (\nu_0 D/\delta_0)$$

which is monic and has algebraic integer coefficients.

Our root bounds provide us with an upper bound on the size of the roots of $Q(X)$: the size of any root of $Q(X)$ is bounded by

$$u = \Xi((\nu_d D/\delta_d)(\nu_{d-1} D/\delta_{d-1}), \ldots, (\nu_d D/\delta_d)^{d-1} (\nu_1 D/\delta_1), (\nu_d D/\delta_d)^d (\nu_0 D/\delta_0)).$$

Since the roots of P are simply the roots of Q divided by $\nu_d D/\delta_d$, this suggests to extend the definitions of u and l as follows:

For an expression E denoting a root of a polynomial of degree d with coefficients given by $E_d, E_{d-1}, E_{d-2}, \ldots, E_0$ we define

$$u(E) = \Xi(\ldots, \left(u(E_d) \prod_{k \neq d} l(E_k)\right)^{d-i} u(E_i) \prod_{k \neq i} l(E_k), \ldots)$$

and

$$l(E) = u(E_d) \prod_{k \neq d} l(E_k).$$

We still need to define the weight $D(E)$ of an expression. We do so in the obvious way. The weight $D(E)$ of an expression E dag is the product of the weights of the nodes and leaves of the dag. Leaves and $+$, $-$, $*$ and $/$-operations have weight 1, a $\sqrt[k]{\ }$-node has weight k, and a $\diamond(j, E_d, \dots)$-operation has weight d.

We can now state our main theorem.

Theorem 1 *Let E be an expression with integer operands and operations $+, -, *, \sqrt[k]{\ }$ for integral k and $\diamond(j, \dots)$ operations. Let ξ be the value of E. Let $u(E)$ and $l(E)$ be defined inductively on the structure of E according to the following rules:*

	$u(E)$	$l(E)$
integer N	$\lvert N \rvert$	1
$E_1 \pm E_2$	$u(E_1) \cdot l(E_2) + l(E_1) \cdot u(E_2)$	$l(E_1) \cdot l(E_2)$
$E_1 \cdot E_2$	$u(E_1) \cdot u(E_2)$	$l(E_1) \cdot l(E_2)$
E_1 / E_2	$u(E_1) \cdot l(E_2)$	$l(E_1) \cdot u(E_2)$
$\sqrt[k]{E_1}$	$\sqrt[k]{u(E_1) l(E_1)^{k-1}}$	$l(E_1)$
$\diamond(j, E_d, \dots, E_0)$	$\Xi(\dots, \left(l(E)^{d-i} u(E_i) \prod_{k \neq i} l(E_k)\right), \dots)$	$u(E_d) \prod_{k \neq d} l(E_k)$

Let $D(E)$ be the weight of E. Then either $\xi = 0$ or

$$\left(l(E) u(E)^{D(E)-1}\right)^{-1} \leq |\xi| \leq u(E) l(E)^{D(E)-1}$$

Proof: We show that the rules for u and l keep the invariant that there are algebraic integers β and γ such that $\xi = \beta/\gamma$ and $u(E)$ is an upper bound on the absolute size of the conjugates of β and $l(E)$ is an upper bound on the absolute size of the conjugates of γ.

We prove this by induction on the structure of E. The base case is trivial. If E is an integer N, we take $\beta = N$ and $\alpha = 1$; β is the root of the polynomial $X - N$ and α is a root of $X - 1$.

Now let $E = E_1 \pm E_2$. By induction hypothesis we have $\xi_j = \beta_j / \gamma_j$ for $j = 1, 2$. We set $\beta = \beta_1 \gamma_2 \pm \beta_2 \gamma_1$ and $\gamma = \gamma_1 \gamma_2$. Since algebraic integers are closed under operations additions, subtractions and multiplications, β and γ are algebraic integers. By Lemma 4, $l(E) = u(E_1) \cdot l(E_2) + l(E_1) \cdot u(E_2)$ is an upper bound on the absolute size of the conjugates of β. Similarly, $l(E)$ is an upper bound on the absolute size of the conjugates of γ.

If $E = E_1 \cdot E_2$, we set $\beta = \beta_1 \beta_2$ and $\gamma = \gamma_1 \gamma_2$. The claim follows analogously to the previous case by Lemma 4.

If $E = E_1 / E_2$, we set $\beta = \beta_1 \gamma_2$ and $\gamma = \beta_2 \gamma_2$. Again, the claim follows using Lemma 4.

If $E = \sqrt[k]{E_1}$, we set $\beta = \sqrt[k]{\beta_1 \gamma_1^{k-1}}$ and $\gamma = \gamma_1$. Since algebraic integers are closed under $\sqrt[k]{\ }$-operations, β is an algebraic integer. By Lemma 4, $u(E)$ is an upper bound on the absolute size of the conjugates of β. There is nothing to show for $\gamma = \gamma_1$.

Finally, let E be defined by a $\diamond(j, E_d, \ldots, E_0)$-operation. We set

$$\beta = \diamond(j, 1, \beta_{d-1}\gamma_d\gamma_{d-2}\cdots\gamma_0, \gamma\beta_{d-2}\gamma_d\gamma_{d-1}\gamma_{d-2}\cdots\gamma_0, \ldots, \gamma^{n-1}\gamma_d\gamma_{d-1}\gamma_{d-2}\cdots\gamma_1\beta_0)$$

and

$$\gamma = \beta_d\gamma_{d-1}\gamma_{d-2}\cdots\gamma_0.$$

By the discussion preceding the statement of our main theorem, $\xi = \beta/\gamma$, β and γ are algebraic integers, $l(E)$ is an upper bound on the absolute size of the conjugates of γ, and $u(E)$ is an upper bound on the absolute value of the conjugates of β. This completes the induction step.

Rewriting ξ as β/γ corresponds to a restructuring of the expression dag defining E into an expression dag E' with a single division-operation. We have $D(E') = D(E)$.

We still need to argue that $D(E)$ is an upper bound on the algebraic degree of β. This follows from the fact that every operation leads to a field extensions whose degree is bounded by the weight of the operation.

We now have collected all ingredients to bound the absolute value of ξ from below. If $\xi \neq 0$, we have $\beta \neq 0$. The absolute value of β and all its conjugates is bounded by $u(E)$. Thus $|\beta| \geq (u(E)^{\deg(\beta)-1})^{-1}$ by Lemma 2. Also $|\gamma| \leq l(E)$. Thus

$$|\xi| = \frac{\beta}{\gamma} \geq \frac{1}{u(E)^{\deg(\beta)-1}} \cdot \frac{1}{l(E)} \geq \frac{1}{u(E)^{D(E)-1} \cdot l(E)}$$

∎

5 Comparison to Other Constructive Root Bounds

The Li-Yap bound is incomparable to the original BFMS bound as well as to our new bound. Its dependence on the degree bound is at least quadratic in the worst case. This dependence occurs for continued fraction type expressions where additions and divisions alternate. It would be very interesting to compare the bounds experimentally. It would also be very interesting to combine the bounds.

In [6] we compared the original BFMS-bound to bounds provided by Mignotte [31] and Canny [12] and showed that the BFMS-bound is never worse than the other bounds for the expressions defined by (1) and (2). Canny's bound also applies to the expressions defined by (1), (2), and (3). We conjecture that the bound presented in this paper is usually better.

6 Testing Wellformedness and Solving Polynomial Systems

In this section, we want to show that the sign test for algebraic expressions described in the introduction also allows us to test whether the value of an expression is well-defined and whether a system of polynomial equations has a solution. We leave it for future work to clarify whether the strategies outlined in this section are of any computational value.

The value of an algebraic expression may be undefined. Divisions by zero and taking a root of even degree of a negative number are easily caught by the sign test. We want to show that the sign test also allows us to test whether the diamond-operation is well defined. For this matter, we need to determine the number of zeros of a polynomial. Sturm sequences, see [31, chapter 5] or [38, Chapter 7] are the appropriate tool. The computation of Sturm sequences amounts to a gcd computation between a polynomial and its derivative. Our sign test is sufficient to implement a gcd computation.

The sign test can also be used to compute the real zeros of a finitely solvable system of polynomial equations in triangular form [32, 20],, i.e., a system of the form

$$\begin{array}{rcl}
f_{(1,1)}(x_1) & = & 0 \\
& \vdots & \\
f_{(1,k_1)}(x_1) & = & 0 \\
f_{(2,1)}(x_1,x_2) & = & 0 \\
& \vdots & \\
f_{(2,k_2)}(x_1,x_2) & = & 0 \\
& \vdots & \\
& \vdots & \\
f_{(n-1,1)}(x_1,\dots,x_{n-1}) & = & 0 \\
& \vdots & \\
f_{(n-1,k_{n-1})}(x_1,\dots,x_{n-1}) & = & 0 \\
f_{(n,1)}(x_1,\dots,x_{n-1},x_n) & = & 0 \\
& \vdots & \\
f_{(n,k_n)}(x_1,\dots,x_{n-1},x_n) & = & 0
\end{array}$$

with $f_{(i,j)} \in \mathbb{Z}[x_1,\dots,x_i] \setminus \mathbb{Z}[x_1,\dots,x_{i-1}]$, $1 \le j \le k_i$. The algorithm is straightforward. One first determines the number of roots of of $f_{(1,1)}$ by means of Sturm sequences. Then one uses the diamond operation to form an expression for each root and feeds the roots into the polynomials $f_{(1,2)},\dots,f_{(1,k_1)}$. One uses the sign test to find out the common roots. Once the common roots of the polynomials in x_1 are known, the same procedure is applied to the block of polynomials in x_1 and x_2 for every common root of the first block. Continuing in this way, we obtain all common roots.

The algorithm extends to arbitrary systems as any finitely solvable system of polynomial equations can be turned into an equivalent system in triangular form [37, 16, 20] by computing a Gröbner basis [3] with respect to a purely lexicographic ordering.

7 Conclusions

The presented bound is the first constructive root bound that guarantees linear dependence on the degree bound for all types of expressions. The work rises several interesting question. How does our bound compare to other bounds, theoretically and experimentally? Is the application to (tridiagonal) systems of polynomials of any computational value?

Acknowledgement: We want to thank Susanne Schmitt for helpful comments.

References

[1] F. Avnaim, J.-D. Boissonnat, O. Devillers, F. Preparata, and M. Yvinec. Evaluating signs of determinants using single-precision arithmetic. *Algorithmica*, 17:111–132, 1997.

[2] H. Brönnimann and M. Yvinec. Efficient exact evaluation of signs of determinants. Research Report 3140, INRIA, 1997.

[3] B. Buchberger. Ein algorithmisches Kriterium für die Lösbarkeit eines algebraischen Geichungssystems. *Aequationes Mathematicae* 4:374-383, 1970.

[4] C. Burnikel. *Exact Computation of Voronoi Diagrams and Line Segment Intersections*. Ph.D thesis, Universität des Saarlandes, March 1996.

[5] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. Efficient exact geometric computation made easy. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 341–350, 1999.

[6] C. Burnikel, R. Fleischer, K. Mehlhorn, and S. Schirra. A strong and easily computable separation bound for arithmetic expressions involving radicals. *Algorithmica* 27:87-99, 2000.

[7] C. Burnikel, K. Mehlhorn, and S. Schirra. How to Compute the Voronoi Diagram of Line Segments: Theoretical and Experimental Results. Proc. 2nd Annu. European Sympos. Algorithms (ESA94), Lecture Notes Comput. Sci. Vol. 855, Springer-Verlag, pages 227-239, 1994.

[8] C. Burnikel, K. Mehlhorn, and S. Schirra. The LEDA class `real` number. Research Report MPI-I-96-1-001, Max-Planck-Institut für Informatik, 1996. A more recent documentation of the implementation is available at `http://www.mpi-sb.mpg.de/~burnikel/reports/real.ps.gz`.

[9] C. Burnikel, S. Funke, and M. Seel. Exact geometric predicates using cascaded computation. In *Proc. 14th Annu. ACM Sympos. Comput. Geom.*, pages 175–183, 1998.

[10] C. Burnikel, J. Könnemann, K. Mehlhorn, S. Näher, S. Schirra, and C. Uhrig. Exact geometric computation in LEDA. In *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pages C18–C19, 1995.

[11] J. Chang and V. Milenkovic. An experiment using LN for exact geometric computations. In *Proc. 5th Canad. Conf. Comput. Geom.*, pages 67–72, 1993.

[12] J.F. Canny. *The Complexity of Robot Motion Planning*. The MIT Press, 1987.

[13] Olivier Devillers, Alexandra Fronville, Bernard Mourrain, and Monique Teillaud. Exact predicates for circle arcs arrangements. In *Proc. 16th Annu. ACM Sympos. Comput. Geom.*, 2000.

[14] T. Dubé, C.K. Yap, *A Basis for Implementing Exact Geometric Algorithms*. October, 1993.

[15] S. Fortune. Progress in computational geometry. In R. Martin, editor, *Directions in Geometric Computing*, pages 81 – 128. Information Geometers Ltd., 1993.

[16] P. Gianni and T. Mora. Algebraic solution of systems of polynomial equations using Groebner bases. Lecture Notes Comput. Sci. Vol. 356, Springer-Verlag, pages 247-257, 1987.

[17] E. Hecke. Vorlesungen über die Theorie der algebraischen Zahlen, 2nd edition, Chelsea, New York, 1970

[18] C. Hoffmann. The problem of accuracy and robustness in geometric computation. *IEEE Computer*, pages 31–41, March 1989.

[19] J.W. Hong. Proving by Example and Gap Theorem. In *Symposium on Foundations of Computer Science*, 1987.

[20] C. Jäger and D. Ratz. A combined method for enclosing all solutions of nonlinear systems of polynomial equations. *Reliable Computing* 1:41-64, 1995.

[21] V. Karamcheti, C. Li, I. Pechtanski, and C. Yap. A core library for robust numeric and geometric computation. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 351–359, 1999.

[22] LEDA. `http://www.mpi-sb.mpg.de/LEDA/leda.html`.

[23] R. Loos. Computing in Algebraic Extensions. Computer Algebra, pages 173 – 187, Springer Verlag, Berlin, 1982

[24] C. Li, and C. K. Yap. A new constructive root bound for algebraic expressions. Manuscript, July 2000.

[25] Z. Li and V. Milenkovic. Constructing strongly convex hulls using exact or rounded arithmetic. *Algorithmica*, 8:345–364, 1992.

[26] G. Liotta, F.P. Preparata, and R. Tamassia. Robust Proximity Queries in Implicit Voronoi Diagrams. Technical Report RI 02912-1910, Center for Geometric Computing, Department of Computer Science, Brown University, May 1996.

[27] K. Mehlhorn and S. Näher. The implementation of geometric algorithms. In *Proceedings of the 13th IFIP World Computer Congress*, volume 1, pages 223–231. Elsevier Science B.V. North-Holland, Amsterdam, 1994.

[28] K. Mehlhorn and S. Näher. *LEDA – A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, England, 1999.

[29] K. Mehlhorn and S. Schirra. Exact Computation with leda_real - Theory and Geometric Applications. Proceedings of the Dagstuhl Seminar *Symbolic-algebraic methods and verification methods- theory and applications* To appear Springer Mathematics series by Springer-Verlag, Wien-New York. Fall 2000.

[30] M. Mignotte. Identification of Algebraic Numbers. *Journal of Algorithms*, 3(3), September 1982.

[31] M. Mignotte. *Mathematics for Computer Algebra*. Springer Verlag, 1992.

[32] B. Mishra. *Algorithmic Algebra*, Springer-Verlag, New York, 1993.

[33] V. J. Milenkovic. Practical methods for set operations on polygons using exact arithmetic. In *Proc. 7th Canad. Conf. Comput. Geom.*, pages 55–60, 1995.

[34] J. Neukirch. Algebraische Zahlentheorie, Springer-Verlag, 1990. volume = ”6”,

[35] E. R. Scheinerman. When close enough is close enough. *American Mathematical Monthly*, 107:489-499, 2000.

[36] S. Schirra. Robustness and Precision Issues in Geometric Computation. Handbook of Computational Geometry, Elsevier, 2000, pages 597-632.

[37] W. Trinks. Über B. Buchbergers Verfahren, Systeme algebraischer Gleichungen zu lösen, J. Number Theory 10:475-488, 1978.

[38] C. K. Yap. *Fundamental problems in algorithmic algebra.* Oxford University Press, 2000.

[39] C. K. Yap. Robust geometric computation. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 653–668, CRC Press, 1997.

[40] C. K. Yap. Towards exact geometric computation. *Comput. Geom. Theory Appl.*, 7:3–23, 1997.

[41] C. K. Yap and T. Dubé. The exact computation paradigm. In D.-Z. Du and F. K. Hwang, editors, *Computing in Euclidean Geometry*, volume 1 of *Lecture Notes Series on Computing*, pages 452–492. World Scientific, Singapore, 2nd edition, 1995.