# An Exact Real Algebraic Arithmetic with Equality Determination

Namhyun Hur,* James H. Davenport
Department of Mathematical Sciences,
University of Bath,
Bath BA2 7AY,
UK
mapnh@maths.bath.ac.uk, jhd@maths.bath.ac.uk.

## ABSTRACT

We describe a new arithmetic model for real algebraic numbers with an exact equality determination. The model represents a real algebraic number as a pair of an arbitrary precision numerical value and a symbolic expression. For the numerical part we currently (another representation could be used) use the dyadic exact real number and for the symbolic part we use a square-free polynomial for the real algebraic number. In this model we show that we can decide exactly the equality of real algebraic numbers.

## 1. INTRODUCTION

We can now calculate not only real algebraic numbers but also some transcendental numbers (more precisely real numbers that are computable)[1] exactly up to any precision as we want. This means that we can generate, for example, decimal digits of a real number to as much precision as we want and we are guaranteed that this representation is accurate to given precision. This contrasts with the situation in most Computer Algebra (CA) systems, where *bigfloats* evaluate to a pre-determined precision. Attempts to produce *lazy bigfloats* in the style of the successful lazy power series have not worked well [2]. The two notable models for exact real arithmetic are : the $B$-adic exact real arithmetic (we choose $B = 2$ hence dyadic) [10] and the linear fractional transformation arithmetic [12]. But these two models suffer from one fundamental problem, namely these models can not tell when a real number is exactly zero [13] or equivalently they cannot tell whether two real numbers are exactly equal or not.

Traditionally the zero identification (or recognition) problem has been investigated mainly by computer algebra community and mostly by Richardson [11]. Although Richardson's method can also be applied to transcendental numbers it uses a combination of difficult mathematics such as the LLL algorithm [7] and Wu's method for generating characteristic sets and we do not know, for example, its complexity. These motivated us to ask whether we can solve the equality problem (hence zero problem) for the smaller set of real algebraic numbers. A real algebraic number, say the positive[2] square root of 2 ($\sqrt{2}$), has two kinds of information associated with it : a numerical value of $1.4142\ldots$ when expanded into decimal and a symbolic expression $x^2 - 2$ since it is a root of this polynomial. Hence we solve the equality problem for $\mathbb{R}_A$ by representing an $\alpha \in \mathbb{R}_A$ as a pair of its numerical value and the minimal polynomial (or possibly a multiple of it) corresponding to it. But the main difference from other models for algebraic number arithmetic such as [8] and [14] is that we use exact real arithmetic whereas others mostly use floating point interval arithmetics. So we use a dyadic real number (we will write $x \in \mathbb{R}_C^{DR}$ to say that $x$ is a dyadic real number) for the numerical part and a square-free (or possibly the minimal polynomial) with integer cofficients as the symbolic part. For example, $\sqrt{2}$ is represented in our model as

$$[f_{\sqrt{2}}(n), x^2 - 2]$$

where the numerical part, $f_{\sqrt{2}}(n)$ is the positive square root of 2 in the dyadic real arithmetic (that is given $n$, $f_{\sqrt{2}}(n)$ returns $n$ bits of exact approximation to $\sqrt{2}$, see next section), and the symbolic part, $x^2 - 2$, is a defining, possibly minimal, polynomial which has the numerical part as one of its real roots.

The motivation for representing a real number as such a pair comes from an observation that we can use the real root separation bound (calculated dyadically), i.e. the minimum distance between any two distinct real roots of a polynomial $p(x) \in \mathbb{Z}[x]$, as a termination condition for the possibly infinite dyadic equality. For the real root separation bound we use Mahler's [3] separation bound for any two real roots of a given polynomial.

*The first author was partially supported during the writing of this paper by the OpenMath project (Esprit project 24969).

[1]We denote the set of real numbers by $\mathbb{R}$, the set of computable real numbers by $\mathbb{R}_C$ and the set of real algebraic numbers by $\mathbb{R}_A$. Note that $\mathbb{R}_A \subset \mathbb{R}_C \subset \mathbb{R}$.


*ISSAC 2000,* St. Andrews, Scotland
©2000 ACM 1-58113-218-2/ 00/ 0008 $5.00

[2]In this paper, $\sqrt{\ }$ always means the positive square root.

To avoid confusion between the dyadic equality (inequality) and pair equality (inequality) we write $x =_{(n)} y$ ($x <_{(n)} y$) to say that $x$ and $y$ are dyadically equal (inequal)[3] and we will use the symbol $=_p$ ($x <_p y$) for our pair equality (inequality) and reserve the standard equality symbol $=$ ($<$) for mathematical equality (inequality) as in $\sqrt{2} \times \sqrt{3} = \sqrt{6}$. We will also use the symbol $:=$ for definition to avoid the confusion between definition and equation.

This paper is organised as follows. In section 2, we briefly describe the dyadic real arithmetic. The reader should note that the choice of dyadic real arithmetic is not fundamental to the argument of this paper: any arbitrary-precision real arithmetic (e.g. [12]) would do. In section 3 we define our pair model for $\mathbb{R}_A$ and the usual elementary operations: addition, subtraction, multiplication, and division. In section 4 we derive the key *Equality Theorem* using Mahler's real root separation bound. We use the theorem to show the equality of two examples. In section 5 we derive the *Inequality Theorem* using the Equality Theorem and show an example of inequality. We conclude by pointing out further points to study and discuss implementation and applications.

## 2. DYADIC EXACT REAL ARITHMETIC

The dyadic exact real arithmetic is an exact real arithmetic based on the concept of computable real numbers and their finite representations. Since the real numbers are uncountable it is obvious that there are only a limited number of real numbers that are finitely representable. These finitely representable real numbers are often called recursive or computable real numbers ($\mathbb{R}_C$). The dyadic real arithmetic represents a number $x \in \mathbb{R}_C$ by a recursive function $f_x : \mathbb{N} \to \mathbb{Z}$ such that

$$|f_x(n) - 2^n x| < 1.$$

Having characterised the set of computable real numbers one can define all the usual operations on them including many transcendental functions [10, 5]. For example, addition can be defined as below which satisfies the bound condition above[4]:

$$f_{x+y}(n) := \left\lfloor \frac{f_x(n+2) + f_y(n+2)}{4} \right\rfloor.$$

But as we said in the introduction one can not have exact equality and inequality due to the fundamental fact that 0 is not decidable (equivalently equality is undecidable) in this arithmetic (the alternative, linear fractional transformational approach, is no different here). So one has to be content with the dyadic equality (and inequality) instead of exact equality (and inequality). But in this paper we show that we can use the dyadic equality (and inequality) for our exact equality (and inequality) determination for $\mathbb{R}_A$.

## 3. $\mathbb{R}_A$ AS PAIRS

We represent an $x \in \mathbb{R}_A$ by a pair of its dyadic real $f_x(n) \in \mathbb{R}_C^{DR}$ and corresponding polynomial $p(x) \in \mathbb{Z}[x]$. Thus in our pair model :

[3]see section 4 for definitions of dyadic (in)equality.

[4]Note that we use an integer division which *rounds* to the nearest integer so that the error is at most 1/2.

**Definition 3.1 (Pair : Squarefree Version)** A real algebraic number in $\mathbb{R}_A^{\text{Pair}}$ is a pair $[f_x(n), p(x)]$ where $f_x(n) \in \mathbb{R}_C^{DR}$ is a dyadic representation of $x$ and $p(x)$ is a square-free polynomial $\in \mathbb{Z}[x]$ such that $p(x) = 0$.

We choose to insist on square-freeness since it is cheap to calculate and necessary to make the discriminant nonzero in the bound used in Theorem 4.5. If we insist that our polynomial be minimal we can define a *minimal algebraic number*. A minimal polynomial is irreducible by definition.

**Definition 3.2 (Pair : Minimal Version)** A minimal real algebraic number is a pair $[f_x(n), p(x)]$ where $f_x(n) \in \mathbb{R}_C^{DR}$ is a dyadic representation of $x$ and $p(x)$ is the minimal polynomial $\in \mathbb{Z}[x]$ such that $p(x) = 0$.

We will write $\bar{x}$ to denote the pair representation of $x \in \mathbb{R}_A$. For example, an integer $k \in \mathbb{R}_A$ is represented by $\bar{k} = [f_k(n), x - k]$ where $f_k(n)$ is a dyadic representation of $k$ (in fact $f_k(n) := 2^n k$) and $x - k$ is the polynomial corresponding to $k$.

### 3.1 Elementary Operations

The elementary operations on pairs follow straightforwardly[5] from the arithmetic of dyadic numbers and the algebra of defining polynomials.

If $\bar{x} =_p [f_x(n), p(x)]$ and $\bar{y} =_p [g_y(n), q(y)]$ are two real numbers in our model, then the pair operations (denoted by $\oplus, \ominus, \otimes$ and $\oslash$) are defined as below. Note that $\hat{+}, \hat{-}, \hat{\times}$ and $\hat{/}$ denote the operations of dyadic real arithmetic and res denotes *resultant* [3]. The resultant calculations do not necessarily give minimal or even square-free polynomials so we need a *refinement* operation which we denoted as $R$.

**Definition 3.3** ($\oplus, \ominus, \otimes, \oslash$)

$$\begin{aligned}
\bar{x} \oplus \bar{y} &:= [f_x(n) \hat{+} g_y(n), \\
&\quad R(\text{res}(\text{res}(z - (x + y), p(x), x), q(y), y))], \\
\bar{x} \ominus \bar{y} &:= [f_x(n) \hat{-} g_y(n), \\
&\quad R(\text{res}(\text{res}(z - (x - y), p(x), x), q(y), y))], \\
\bar{x} \otimes \bar{y} &:= [f_x(n) \hat{\times} g_y(n), \\
&\quad R(\text{res}(\text{res}(z - (x \times y), p(x), x), q(y), y))], \\
\bar{x} \oslash \bar{y} &:= [f_x(n) \hat{/} g_y(n), \\
&\quad R(\text{res}(\text{res}((y \times z) - x), p(x), x), q(y), y))].
\end{aligned}$$

The resultant calculation in the symbolic part returns a polynomial which has the corresponding numerical part as one of its roots. Note that we will have to take care of the cases where the resulting polynomials are not square-free. In these cases we square-free-decompose them into products of square-free polynomials. In principle, we can deal with these polynomials by considering their product, i.e. the square-free part of the original, but we decided to simplify the resulting polynomials by choosing the one which contains the

[5]But this does not necessarily mean that it is easy to code them. We implemented our model in Axiom. Axiom is a trademark of NAG Ltd..

numerical part as a root from among the square-free factors. All these matters are taken care of by the refinement operation (Lemma 4.6). Note that the resultant calculation and square-free factorisation are standard parts of almost all algebraic number packages [8]. An interesting contrast between our pair representation and those (minimal polynomial, interval) pair representations is: other arithmetics [8, 14] have to specify *which root* of the polynomial that it means whereas we have to choose *which polynomial* contains the root that is given as the numerical part. This is due to the fact that our model includes the exact numerical information while others include the root-location information in terms of intervals. In choosing the right polynomial we will fully use the numerical information provided by dyadic exact real arithmetic as described below in detail.

Below is a simple example of $\bar{2} \oplus \bar{3}$ in Axiom. Note that $\bar{2} = [f_2(n), x-2]$ and $\bar{3} = [f_3(n), y-3]$. In Axiom's language, `2::PAIR` and `3::PAIR` gives $\bar{2}$ and $\bar{3}$ respectively. Note that we only show five decimal digits of the numerical value for convenience. The `?` is the Axiom's symbol for a variable[6].

```
(1) -> a := 2::PAIR
   (1)  ["+2.00000",? - 2]                Type: PAIR
(2) -> b := 3::PAIR
   (2)  ["+3.00000",? - 3]                Type: PAIR
(3) -> a + b
   (3)  ["+5.00000",? - 5]                Type: PAIR
```

For other representations of algebraic numbers see those references cited above and [9]. These representations seem to differ only in the way how they locate the roots of the polynomials and all seem to rely on the calculation of Sturm sequences to count the number of sign variations.

# 4. AN EXACT EQUALITY FOR $\mathbb{R}_A$

As we remarked earlier we write $x =_n y$ to say that $x$ and $y$ are dyadically equal up to $n$ digits when $x(m) = y(m)$ for all $m \leq n$ and $x <_n y$ to say that $x$ is dyadically smaller than $y$, or more precisely, if we can find an $n$ such that $y(n) \geq x(n) + 2$. Note that the $+2$ is needed, since we have a tolerance of $\pm 1$ on each of $x$ and $y$.

Let $f_{\sqrt{k}}(n)$ be a dyadic square root function such that

$$\left| f_{\sqrt{k}}(n) - 2^n \sqrt{k} \right| < 1$$

for positive integer k. Translated into a pair representation, $\sqrt{2} \times \sqrt{3} = \sqrt{6}$ becomes

$$[f_{\sqrt{2}}(n), x^2 - 2] \otimes [f_{\sqrt{3}}(n), x^2 - 3] =_p [f_{\sqrt{6}}(n), x^2 - 6]$$

and we are claiming that the pair equality is the same as the mathematical equality.

We now derive the Equality Theorem of two real algebraic numbers as represented as pairs. For this we need the following theorem of Mahler [3]. Throughout this section $p(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0$ where the $a_i$ are integers and $a_n$ is non-zero. $n$ is therefore the degree of $p$. The separation of $p$, $\mathrm{sep}(p)$ is defined as the smallest distance between any two roots of $p$. Let the roots of $p$ be $\alpha_1, \ldots, \alpha_n$.

**Definition 4.1** ($\mathrm{sep}(p)$)

$$\mathrm{sep}(p) := \min_{1 \leq i < j \leq n} |\alpha_i - \alpha_j| .$$

Note that $\mathrm{sep}(p) \neq 0$ if, and only if, $p$ is square-free.

We also need the concept of the *discriminant* of a polynomial.

**Definition 4.2** ($\mathrm{disc}(p)$) The *discriminant* of a polynomial $p$, $\mathrm{disc}(p)$, with leading coefficient $a_n$ and roots $\alpha_1, \ldots, \alpha_n$ is defined as

$$\mathrm{disc}(p) := a_n^{2n-2} \prod_{1 \leq i \leq j \leq n} (\alpha_i - \alpha_j)^2.$$

If $a_i$ are all integers then the discriminant is also an integer, nonzero if, and only if, the polynomial is square-free. Mahler gave a bound for the separation [3].

**Theorem 4.3 (Mahler)**

$$\mathrm{sep}(p) > \sqrt{3\,|\mathrm{disc}(p)|}\; n^{-(n+2)/2}\; ||p||_2^{1-n}.$$

*where*

$$||p||_2 := \sqrt{\sum_{i=0}^{n} |a_i|^2}.$$

Theorem 4.3 gives us a lower bound for the minimum distance between any two distinct real roots. We will call it Mahler's bound or in short Mbound.

**Definition 4.4** ($\mathrm{Mbound}(p)$)

$$\mathrm{Mbound}(p) := \sqrt{3\,|\mathrm{disc}(p)|}\; n^{-(n+2)/2}\; ||p||_2^{1-n}$$

Using the Mahler's bound we can derive our key theorem.

**Theorem 4.5 (Equality Theorem)** *Let*

$$\bar{x} =_p [f_x(n), p(x)] \text{ and } \bar{y} =_p [g_y(n), q(y)].$$

*Then*

$$\bar{x} =_p \bar{y} \;\; \textit{iff} \;\; |f_x(n) \mathbin{\hat{-}} g_y(n)|_{(n)} <_{(n)} \mathrm{Mbound}\, r(z)$$

*where $r(z)$ is the symbolic part of $\bar{x} \ominus \bar{y}$.*

PROOF. If we negate Mahler's Theorem then, for any two real algebraic numbers $\bar{x} = [f_x(n), p(x)]$ and $\bar{y} = [g_y(n), q(y)]$, if the (dyadic) distance between the two numerical parts, $|f_x(n) \mathbin{\hat{-}} g_y(n)|_{(n)}$, is (dyadically) less than Mbound of the minimal polynomial corresponding to $\bar{x} \ominus \bar{y}$, then $x$ and $y$ are the same. □

Note that the refinement operation is very important in the sense that it guarantees that $r(z)$ is a square-free polynomial, i.e. $\mathrm{disc}\, r(z) \neq 0$. In the rest of this section we show two examples.

[6] In Definition 3.3, $x$'s polynomial was in terms of $x$, $y$'s in terms of $y$, and the answer's in terms of $z$. In the implementation, we use Axiom's anonymous "SparseUnivariatePolynomial" type.

## 4.1 Example 1 : $\sqrt{2} \times \sqrt{3} = \sqrt{6}$

We apply the Equality Theorem to show $\sqrt{2} \times \sqrt{3} = \sqrt{6}$. In other words we have to show whether

$$[f_{\sqrt{2}}(n), x^2 - 2] \otimes [f_{\sqrt{3}}(n), x^2 - 3] =_p [f_{\sqrt{6}}(n), x^2 - 6].$$

or not. Expanding the left-hand side we get

$$\begin{aligned} & [f_{\sqrt{2}}(n), x^2 - 2] \otimes [f_{\sqrt{3}}(n), x^2 - 3] \\ =_p & [f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}}(n), R(\mathrm{res}(\mathrm{res}(z - (x \times y), p(x), x), q(y), y))] \\ =_p & [f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}}(n), R(x^4 - 12x^2 + 36)] \\ =_p & [f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}}(n), x^2 - 6]. \end{aligned}$$

In Axiom,

```
(1) -> x := sqrt(2::PAIR) * sqrt(3::PAIR)
                      2
   (1)  ["+2.44949",?  - 6]                Type: PAIR
(2) -> y := sqrt(6::PAIR)
                      2
   (2)  ["+2.44949",?  - 6]                Type: PAIR
```

Notice the refinement operation $R$ above performed a square-free decomposition and lowered multiplicity two into one to make the discriminant non-zero. So our problem is now changed to

$$[f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}}(n), x^2 - 6] =_p [f_{\sqrt{6}}(n), x^2 - 6]$$

At this stage we might want to say that they are equal as pairs (hence mathematically equal). But unfortunately we can not insist that they are equal as pairs yet for two reasons: the equality of the symbolic parts is not sufficient for the equality of numerical parts (for example, $\sqrt{2} \times \sqrt{3} \neq -\sqrt{6}$) and the numerical parts are not mathematically equal but dyadically equal. So we apply the Equality Theorem. Applying $\ominus$ we have

$$\begin{aligned} & [f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}}(n),\ x^2 - 6] \ominus [f_{\sqrt{6}}(n),\ x^2 - 6] \\ & \quad =_p [f_{\sqrt{2}} \hat{\times} f_{\sqrt{3}} \hat{-} f_{\sqrt{6}}(n),\ x^2(x^2 - 24)]. \end{aligned}$$

In Axiom (using a version without the refinement operation),

```
(3) -> x-y
                  2
   (3)  ["+0",?(?  - 24)]                  Type: PAIR
```

The polynomial $x(x^2 - 24)$ has three real roots $0, 2\sqrt{6}$ and $-2\sqrt{6}$. Now by the Equality Theorem if

$$|f_0(n)| <_{(n)} \mathrm{Mbound}\,(x^2(x^2 - 24))$$

then they are equal. Since (for $n \geq 5$)

$$f_0(n) <_{(n)} \mathrm{Mbound}\,(x^2 - 24) =_5 0.17662$$

we have shown that they are indeed equal.

## 4.2 Example 2 : $\sqrt{9 + 4\sqrt{2}} = 1 + 2\sqrt{2}$

This example is taken from [4]. The equality corresponds to the pair equality

$$[f_{\sqrt{9+4\sqrt{2}}}(n), x^4 - 18x^2 + 49] =_p [f_{1+2\sqrt{2}}(n), x^2 - 2x - 7].$$

In Axiom,

```
(1) -> x := sqrt(9+4*sqrt(2::PAIR))
                     4      2
   (1)  ["+3.82843",?  - 18?  + 49]        Type: PAIR
```

A factorising refinement would give

```
(2) -> x := sqrt(9+4*sqrt(2::PAIR))
                    2              2
   (2)  ["+3.82843",(?  - 2? - 7)(?  + 2? - 7)]
                                        Type: PAIR
```

and

```
(3) -> y := 1+2*sqrt(2::PAIR)
                      2
   (3)  ["+3.82843",?  - 2? - 7]           Type: PAIR
```

We now have an interesting problem : which one, $x^2 - 2x - 7$ or $x^2 + 2x - 7$, has $\sqrt{9 + 4\sqrt{2}}$ as one of its roots? This question is not so straightforward to answer. Our answer is to use the dyadic inequality as described below. First we evaluate the two factors at $x = \sqrt{9 + 4\sqrt{2}}$. Only one of these two evaluations must return zero since one of them must have the given number as a root and all the roots are distinct. Now we **only have to check which evaluation becomes zero**. But this looks like we ended up at the same problem we first set out to solve. But fortunately we have the following lemma. This lemma will allow us to choose the right one among the factors. In this example we have only two factors but the general case is no harder. We will write defpoly $(x)$ for the square-free defining polynomial for $x$.

**Lemma 4.6 (Refinement Lemma)** *Let $p(x)q(x)$ be a factored square-free polynomial (i.e., $p, q$ are square-free and their gcd is 1) corresponding to a real algebraic number $x$. Then*

$$\mathrm{defpoly}\,(x) = \begin{cases} p(x) & \text{if } |p(x)|_{(n)} <_{(n)} |q(x)|_{(n)} \\ q(x) & \text{if } |q(x)|_{(n)} <_{(n)} |p(x)|_{(n)} \end{cases}$$

PROOF. We know that only one of them is (exactly) zero hence it must be dyadically zero and it is the smaller one (or smallest if there are more than two candidates) and thus it must be the dyadically smaller one. □

To apply the lemma, we check

$$\begin{aligned} & \left|(x^2 - 2x - 7)_{x=_{(n)} f_{\sqrt{9+4\sqrt{2}}}(n)}\right|_{(n)} \\ <_{(n)} & \left|(x^2 + 2x - 7)_{x=_{(n)} f_{1+2\sqrt{2}}(n)}\right|_{(n)} ? \end{aligned}$$

The actual dyadic values are 0 for $x^2-2x-7$ and 15.31371 for the other. So $x^2-2x-7$ is the defining polynomial for $\sqrt{9+4\sqrt{2}}$. Having settled the choice problem we now have

$$[f_{\sqrt{9+4\sqrt{2}}}(n), x^2-2x-7] =_p [f_{1+2\sqrt{2}}(n), x^2-2x-7]$$

From now on it is exactly the same routine as the first example. So taking $\ominus$ we get

$$[f_{\sqrt{9+4\sqrt{2}}}(n) \hat{-} f_{1+2\sqrt{2}}(n), x(x^2-32)].$$

and since the numerical value of $f_{\sqrt{9+4\sqrt{2}}-1+2\sqrt{2}}(n)$ is less than

$$\text{Mbound}(x(x^2-32)) =_5 0.03925,$$

they are indeed equal.

# 5. AN EXACT INEQUALITY FOR $\mathbb{R}_A$

In this section we study inequality. The problem is solved step by step. We use the symbol $<_p$ for pair inequality. First note that we can assert that $\bar{x} <_p \bar{y} = [f_x(n), p(x)] <_p [g_y(n), q(y)]$ (pair inequality) if $f_x(n) <_n g_y(n)$. But unfortunately we have no guarantee for the termination of the dyadic inequality. Indeed they will run forever if they happen to be equal. Thus what we need is a termination condition which will guarantee the pair inequality without resorting to dyadic inequality infinitely[7]. We use the pair equality to derive such a termination condition as below. First, we check whether $\bar{x} =_p \bar{y}$ or not using the pair equality. If yes then obviously $\bar{x} \not<_p \bar{y}$. If no, then we can safely resort to dyadic inequality since we are certain that they will return either yes or no eventually although this might take arbitrarily long time. Hence we have the following theorem.

**Theorem 5.1 (Inequality Theorem)** *Let*

$$\bar{x} =_p [f_x(n), p(x)] \text{ and } \bar{y} =_p [g_y(n), q(y)].$$

*Then*

$$\begin{aligned}
\bar{x} <_p \bar{y} \quad &\text{iff} \quad \bar{x} \neq_p \bar{y} \text{ and } f_x(n) <_n g_y(n),\\
\bar{x} >_p \bar{y} \quad &\text{iff} \quad \bar{x} \neq_p \bar{y} \text{ and } g_y(n) <_n f_x(n),\\
\bar{x} \leq_p \bar{y} \quad &\text{iff} \quad \bar{x} =_p \bar{y} \text{ or } \bar{x} <_p \bar{y},\\
\bar{x} \geq_p \bar{y} \quad &\text{iff} \quad \bar{x} =_p \bar{y} \text{ or } \bar{x} >_p \bar{y}.
\end{aligned}$$

[7]Notice that we used an (dyadic) *inequality* lemma in showing *equality*. For showing *inequality* we have to use *equality* information.

# 6. CONCLUSION AND FURTHER STUDY

In this paper we have shown that we can decide the equality and inequality for $\mathbb{R}_A$. But in general, Rice [13] showed that zero is undecidable. Due to this result we can not, for example, determine the integer part of a real number which, in turn, implies the undecidability of the rationality/irrationality of a real number. But now with our algorithm we can at least test equality for any two real algebraic numbers. Equivalently we can determine whether a real algebraic number is zero or not. So one can regard this as a partial but practical solution for the undecidability of zero problem.

The key result in this process is Theorem 4.5, which relies on the bound in Lemma 4.3. If such a bound could be found for a wider class of expressions, this approach would generalise.

We regret that we couldn't perform a complexity analysis of our algorithms. As far as we know the $B$-adic exact real arithmetic itself lacks any kind of complexity information (the same situation holds for the linear fractional transformation approach).

The next step for the zero recognition problem is to consider the possibility of zero recognition for a larger class of numbers which includes some transcendental numbers. Currently Richardson's method seems the best at the moment. One important result related with this is the model-completeness proof for the first order theory $(\mathbb{R}, exp)$ [15].

Another area related to our work is in denesting nested radicals [6]. It will be interesting if we can incorporate these simplifications into our algorithm so that we can simplify nested radicals first before testing equality.

Several experiments need to be performed to tune this implementation. For example, it would be possible to replace the numeric part of $\sqrt{2}+\sqrt{3}$, whose minimal polynomial is $x^4-10x^2+1$, by a numerical algorithm directly approximating this root (we do know a starting value, generally the major problem in numerical root-finding). Is this worth it?

We have only discussed real arithmetic. Since $\mathbb{C}_A \cong \mathbb{R}_A \times \mathbb{R}_A$, we have a representation of $\mathbb{C}_A$ as $\mathbb{R}_A^{\text{Pair}} \times \mathbb{R}_A^{\text{Pair}}$. But is this the most efficient one? Maybe we should have a triple $(\Re(x) \in \mathbb{R}_C^{DR}, \Im(x) \in \mathbb{R}_C^{DR}, p)$, where $p$ is a polynomial satisfied by $x$.

## 7. REFERENCES


[1] Z. Adamowicz, and P. Zbierski, *Logic of Mathematics: a modern course of classical logic*, John Wiley and Sons, 1997.

[2] H. J. Boehm, R. Cartwright, M. J. O'Donnel, and M. Riggle, *Exact real arithmetic : a case study in higher order programming*, Proceedings of the 1986 ACM Conference on LISP and Functional Programming, ACM, 1986, pp. 162-173.

[3] J. H. Davenport, *Computer algebra for cylindrical algebraic decomposition*, Technical Report 88-10, Department of Mathematical Sciences, University of Bath, Claverton Down, Bath BA2 7AY, England. `http://www.bath.ac.uk/~masjhd/TRITA.dvi`

[4] J. H. Davenport, Y. Siret and E. Tournier, *Computer Algebra*, 2nd edition, Academic Press, 1993.

[5] J. R. Harrison, *Introduction to functional programming*, Available from the web: `http://www.cl.cam.ac.uk/~jrh`.

[6] S. Landau, *How to tangle with a nested radical*, The Mathematical Intelligencer, vol. 16, no. 2, pp. 49-55.

[7] A. K. Lenstra, H. W. Lenstra Jr., and L. Lovász, *Factoring Polynomials with Rational Coefficients*, Math. Ann., 261, 1982, pp. 515-534.

[8] R. Loos, *Computing in algebraic extensions*, Computing, **4**(suppl.), pp. 173-187.

[9] B. Mishra, *Algorithmic Algebra*, Texts and monographs in computer science, Springer-Verlag, 1993.

[10] V. Ménissier-Morain, *Arithméique exacte : conception, algorithmique et performances d'une implméntation informatique en précision arbitraire*, PhD Thesis, Université Paris 7, December 1994.

[11] D. Richardson, *How to recognize zero*, Journal of Symbolic Computation, **24**, 1997, pp. 1-19.

[12] P. J. Potts, *Exact real arithmetic using Möbius transformations*, PhD Thesis, Imperial College, July 1998.

[13] H. G. Rice, *Recursive real numbers*, Proceedings of the American Mathematical Society, **5**, 1954, pp 784-791.

[14] A. W. Strzeboński, *Computing in the field of complex algebraic numbers*, Journal of Symbolic Computation, **24**, 1997, pp. 647-656.

[15] A. J. Wilkie, *Model completeness results for expansions of the ordered field of real numbers by restricted pfaffian functions and the exponential function*, Journal of the AMS, **9**, 1996, pp. 1051-1094.