Construction of real algebraic numbers in Coq

Cyril Cohen

Construction of real algebraic numbers in Coq

Cyril Cohen

INRIA Saclay–Île-de-France,
LIX École Polytechnique
Microsoft Research - INRIA Joint Centre
cohen@crans.org

Abstract. This paper shows a construction in Coq of the set of real algebraic numbers, together with a formal proof that this set has a structure of discrete Archimedean real closed field. This construction hence implements an interface of real closed field. Instances of such an interface immediately enjoy quantifier elimination thanks to a previous work. This work also intends to be a basis for the construction of complex algebraic numbers and to be a reference implementation for the certification of numerous algorithms relying on algebraic numbers in computer algebra.

Introduction

Real algebraic numbers form the countable subset of real numbers which are roots of polynomials with rational coefficients. This strict sub-field of real numbers has interesting properties that make it an important object for algorithms in computer algebra and in constructive and effective mathematics. For example, they can be substituted for real numbers in the ongoing constructive formalization of Feit-Thompson Theorem. Indeed, there is an effective algorithm to compare two algebraic numbers and all field operations can be defined in an exact way. Moreover, they can be equipped with a structure of discrete Archimedean real closed field, which is an Archimedean ordered field with decidable ordering satisfying the intermediate value property for polynomials.

The aim of this paper is to show how we define in Coq a data-type representing the real algebraic numbers and to describe how to formally show it is an Archimedean real closed field. This construction and these proofs are described in many standard references on constructive mathematics [11] or in computer algebra [2]. However, the implementation of these results in a proof assistant requires various changes in their presentation. Hence our development is not a literate translation of a well-chosen reference, but is rather a synthesis of results from the mathematical folklore which are often unused in the literature because they are subsumed by classical results.

In order to define real algebraic numbers, standard references usually suggest one of the following strategies. The first one takes a type representing real numbers and builds the type representing the subset of reals which are roots of a polynomial with rational coefficients. One must then show that induced arithmetic operations on this subset have the expected properties. The second

strategy starts from a type representing rational numbers and formalizes the real closure of rational numbers, which is the smallest real closed field containing them. An element of the closure is usually represented as a pair polynomial - interval, satisfying the invariant that the polynomial has a unique root in the interval. This selected root is the algebraic number encoded by that pair. From a constructive point of view, there is no reason to prefer one or the other of these strategies: it may of course be possible to complete the required proofs in any of these two cases. However, there are significant differences in the nature of objects and proofs we handle when formalized in type theory.

In this work, we combine the two approaches in order to get the advantages of both and to eliminate their respective drawbacks.

Constructive formal libraries on exact reals are available in the Coq system [8]. However, for the requirements of this formalization we developed a short library constructing exact reals as Cauchy sequences from an arbitrary Archimedean field. We explain these formalization choices and our construction in Section 2.

Then, in Section 3 we introduce a first type for algebraic real numbers which we call algebraic Cauchy reals, together with its comparison algorithm and arithmetic operations. In particular, we show how to compute annihilating polynomials, decide the equality and more generally the comparison.

We then describe in Section 4 how to construct the real closure of rational numbers to get a second data-type for real algebraic numbers, that we call real algebraic domain.

Thanks to this second data-type and to the equality decision procedure, we show in Section 5 how to form the real algebraic numbers and we prove that it is a real closed field. The key ingredient is the proof of the intermediate value property for polynomials, which concludes this work.

The complete Coq formalization we describe in this paper is available at `http://perso.crans.org/cohen/work/realalg`. The code excerpts of the paper may diverge from the actual code, for the sake of readability. However, we wrote the proofs in a way which is very close to their Coq formalization.

1 Preliminaries

In this work, we use the SSReflect library of the *Mathematical Components* project [13]. We base our development on the algebraic hierarchy [7], with the extensions we already brought to describe discrete ordered structures [5]. We use mostly the discrete real closed field structure. We also take advantage of the available libraries on polynomials with coefficients in rings or fields. More precisely, we use the polynomial arithmetic library which grants the following definitions and properties: arithmetic operations, euclidean division, Bézout theorem, Gauss theorem.

We explain in more details some elements of the SSReflect library we use.

In the SSReflect library, algebraic structures are equipped with a decidable equality and a choice operator.

Decidable equality structure

Decidable equality structures are instances of an interface called `eqType`. Such a structure is a dependently typed record that bundles a type, together with a boolean relation `(eq_op : T → T → bool)` and a proof it reflects the Leibniz equality, which means:

```
∀ (T : eqType) (x y : T), x = y ↔ (eq_op x y = true)
```

The SSReflect library provides a rich theory about `eqType`, such as for example the uniqueness of equality proofs on such types. The importance of this structure also comes from the SSReflect methodology to go back and forth between boolean statements and propositional statements in order to alternate computational steps with deductive steps.

Choice structure

Choice structures are instances of an interface called `choiceType` in the library. They provide us the choice operator `xchoose` of type:

```
xchoose : ∀ (T : choiceType) (P : T → bool), (∃ x, P x) → T.
```

which satisfies the two following properties :

```
xchooseP : ∀ (T : choiceType) (P : T → bool) (xP : ∃ x, P x),
  P (xchoose T P xP).
eq_xchoose : ∀ (T : choiceType) (P Q : T → bool)
               (xP : ∃ x, P x) (xQ : ∃ x, Q x),
  (∀ x, P x = Q x) → xchoose T P xP = xchoose T Q xQ.
```

which respectively ensure the correctness and uniqueness of the chosen element with respect to the predicate `P`.

For instance, in Coq, any countable type can be provably equipped with such a structure. This means we can take T to be the type $\mathbb{Q}$ of rational numbers.

The choice structure is fundamental to formalize both the comparison of Cauchy reals in Section 2.2 and the construction of the effective quotient type in Section 5.

Resultant of two polynomials and corollary to Bézout theorem

The resultant of two polynomials $P = \sum_{i=0}^{m} p_i X^i$ et $Q = \sum_{i=0}^{n} q_i X^i$ is usually defined as the determinant of the Sylvester matrix.

$$\mathrm{Res}_X(P,Q) = \begin{vmatrix} p_m & p_{m-1} & \cdots & p_0 & 0 & 0 & \cdots & 0 \\ 0 & p_m & p_{m-1} & \cdots & p_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & p_m & p_{m-1} & \cdots & p_0 & 0 \\ 0 & \cdots & 0 & 0 & p_m & p_{m-1} & \cdots & p_0 \\ q_n & q_{m-1} & \cdots & q_0 & 0 & 0 & \cdots & 0 \\ 0 & q_n & q_{m-1} & \cdots & q_0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & q_n & q_{m-1} & \cdots & q_0 & 0 \\ 0 & \cdots & 0 & 0 & q_n & q_{m-1} & \cdots & q_0 \end{vmatrix}$$

The notion of resultant is well described and studied in numerous books, we invite the reader to look in one of them, for instance in [10]. If the polynomials are univariate the resultant is a scalar, if they are bivariate, it is a univariate polynomial in the remaining variable. In our development we use only the two following properties of the resultant $\mathrm{Res}_X(P(X,Y),Q(X,Y)) \in F[Y]$ of polynomials $P, Q \in F[X,Y]$ where F is a field:

$$\exists U, V \in F[X,Y], \quad \mathrm{Res}_X(P,Q) = UP + VQ$$
$$\mathrm{Res}_X(P,Q) = 0 \quad \Leftrightarrow \quad P \text{ and } Q \text{ are not coprime as polynomials in } X$$

which respectively express that the resultant of P and Q is in the ideal generated by P and Q, and is zero if and only if P and Q are not coprime as polynomials in X with coefficients in $F[Y]$, i.e. they have no common factor in $(F[Y])[X]$.

Moreover we use the following corollary to Bézout theorem: If P and Q are not coprime as polynomials in X with coefficients in $F[Y]$, there exist U and V in $F[X,Y]$ such that U is non zero, $\deg_X(U) < \deg(Q)$ and

$$U(X,Y)P(X,Y) = V(X,Y)Q(X,Y)$$

2 Construction and properties of Cauchy reals

From now on, we denote by F an ordered Archimedean field equipped with a decidable equality structure and with a choice structure. All the constructions are done over F which is, for our purpose, an appropriate generalization of $\mathbb{Q}$. Although it is necessary for this construction, we do not detail the use of the Archimedean property for the sake of readability.

It remains unclear whether an axiomatization of Cauchy reals as described in [8] would fit our needs. Moreover, our implementation is shorter and more direct, but less generic, when compared with Russell O'Connor's [12].

2.1 Mathematical description and Coq data-type

We define a Cauchy real as a sequence $(x_n)_{n\in\mathbb{N}}$ in $F^{\mathbb{N}}$, together with a convergence modulus $m_x : F \to \mathbb{N}$ such that from the index $m_x(\varepsilon)$, the distance between any two elements is smaller than ε. This "Cauchy property" is stated as:

$$\forall \varepsilon\ \forall i\ \forall j\ ,\ m_x(\varepsilon) \leq i\ \wedge\ m_x(\varepsilon) \leq j\ \Rightarrow\ |x_i - x_j| < \varepsilon$$

We encode sequences of elements of F as functions from natural numbers to F. Hence, we encode Cauchy reals by packaging together the sequence $(x_n)_n$, the modulus m_x and the "Cauchy property":

```
Definition creal_axiom (x : nat →F) :=
  {m : F →nat | ∀ε i j, m ε ≤ i →m ε ≤ j →|x i - x j| < ε}.
Inductive creal := CReal
  {cauchyseq : (nat →F);      _ : creal_axiom cauchyseq}.
```

We remind that `{m : F →nat | ...}` is called a sigma-type and can be read "there exist a function `(m : F →nat)` such that".

The C-CoRN library also provides an interface for Cauchy reals and a construction of Cauchy sequences, which is used to instantiate the interface in [8]. Although their definition is close enough to ours, we redefine and re-implement Cauchy reals from scratch, mainly because our algebraic structures are incompatible. We use this as an opportunity to restate the definitions in a way which is more compatible with our proof style.

In this paper, we often denote a Cauchy sequence $(x_n)_n$ of convergence modulus m_x by the notation $\bar{x}$. We call such an element a Cauchy real and it represents a constructive real number. We often take the ith element of the underlying Cauchy sequence of $\bar{x}$, and we denote it as x_i. Moreover, in Coq code, m_x is encoded as a function `(cauchymod x)` of type `(F →nat)`. A Coq user will remark that such a function is definable because the existential modulus in the definition of the Cauchy sequence is in `Type`.

By definition of Cauchy sequences, we get the following property:

```
Lemma cauchymodP (x : creal) (ε : F) (i j : nat) :
  cauchymod x ε ≤ i →cauchymod x ε ≤ j →|x i - x j| < ε
```

It is important to note that when we apply this lemma, we produce a sub-goal (which we call side condition) of the form $f(\varepsilon) \leq i$. This is a general scheme in our development: during a proof we may generate n side conditions $f_k(\varepsilon) \leq i$ for $k \in \{1, \ldots, n\}$. Indeed, if all constraints on i are formulated like this, it suffices to take i to be the maximum of all the $f_k(\varepsilon)$, in order to satisfy all the side conditions on i. We even have designed an automated procedure to solve this kind of constraints using the `Ltac` language [6] available in Coq, so that many proofs begin with a command meaning "let i be a big enough natural number".

From `cauchymod` we can define a function `ubound` to bound above the values of elements of a Cauchy sequence. It then satisfies the following property:

```
Lemma uboundP : ∀(x : creal) (i : nat), |x i| ≤ ubound x.
```

In the rest of the development, this function is used to compute the convergence moduli of numerous Cauchy sequences. We use the notation $\lceil x \rceil$ for `(ubound x)`.

2.2 Comparison

On Cauchy reals, the Leibniz equality is not a good notion to compare numbers, as two distinct sequences may represent the same real number. In fact, the good correct of equality on Cauchy reals states that $\bar{x}$ and $\bar{y}$ are equivalent if the sequence of point-wise distances $(|x_n - y_n|)_n$ converges to 0.

A type together with an equivalence relation is called a setoid, and the equivalence is the setoid equality. Coq provides tools to declare setoids, to declare functions that are compatible with the setoid equality, and eventually to rewrite using the setoid equality in contexts that are compatible with it [1].

Although the comparison of Cauchy reals is not decidable, telling whether $\bar{x}$ and $\bar{y}$ are distinct is semi-decidable: classically, if they are not equal, there

exist a quantity δ and an index k such that $\delta \leq |x_i - y_i|$ for all i greater than k. Hence the primitive notion for comparison is not equality but apartness, which contains additional information: a witness for the non-negative lower bound of the gap separating the two sequences.

For the sake of clarity we write $\bar{x} \neq \bar{y}$ for apartness and $\bar{x} \equiv \bar{y}$ for its negation. The notion of non apartness coincides with the notion of equivalence stated above and is declared as the setoid equality on Cauchy reals.

From a proof of apartness $\bar{x} \neq \bar{y}$ we must be able to extract a rank k and a non-negative witness δ which bounds below the sequence $(|x_n - y_n|)_n$ from the rank k. This lower bound is needed to define the inverse as described in Section 2.4. So we could define apartness as follows, using a witness in `Type` to make it available for computation:

```
Definition bad_neq_creal x y : Type := {δ : F | 0 < δ &
∀i, cauchymod x δ ≤ i →cauchymod y δ ≤ i →δ ≤ |x i - y i|}.
```

But to be fully compatible with the setoid mechanism, the apartness must be in `Prop`, not in `Type`. Robbert Krebbers and Bas Spitters [9] already encountered this problem in C-CoRN and solved it using the "constructive indefinite description" theorem, which is provable for decidable properties whose domain is `nat`. Our solution uses a variant of this theorem, thanks to the `choiceType` structure of `F`.

We define apartness ($\neq$) as follows:

```
Definition neq_creal (x y : creal) : Prop :=
  ∃δ, (0<δ) && (3 * δ ≤ |x (cauchymod x δ) - y (cauchymod y δ)|).
```

Then, using `xchoose`, we can define the non-negative lower bound function:

```
Definition lbound x y (neq_xy : x ≠y) : F := xchoose F _ neq_xy.
```

Given two Cauchy reals $\bar{x}$ and $\bar{y}$ which are provably apart from each other, let δ be their non-negative lower bound of separation as defined above. From `xchooseP` we get that $3\delta \leq \left|x_{m_x(\delta)} - y_{m_y(\delta)}\right|$. Thus:

$$\forall i, \quad 3\delta \leq \left|x_{m_x(\delta)} - x_i\right| + |x_i - y_i| + \left|y_i - y_{m_y(\delta)}\right|$$

But since we work on Cauchy sequences, we know how to bound the distance between any two elements of the sequence, starting from a well chosen index: $\forall\, i \geq m_x(\delta),\ \left|x_{m_x(\delta)} - x_i\right| < \delta$ and $\forall\, i \geq m_y(\delta),\ \left|y_i - y_{m_y(\delta)}\right| < \delta$. So:

$$\forall\, i \geq \max(m_x(\delta), m_y(\delta)), \quad \delta \leq |x_i - y_i|$$

Hence we prove the lemma:

```
Lemma lboundP (x y : creal) (neq_xy : x ≠y) i :
  cauchymod x (lbound neq_xy) ≤ i →
  cauchymod y (lbound neq_xy) ≤ i →lbound neq_xy ≤ |x i - y i|.
```

2.3 Order relation

The order relation is handled the same way as apartness. The primitive notion is the strict ordering, the negation of which defines the non-strict ordering. For the sake of space we don't write much about comparison as beyond noting it is derivable from a proof of apartness:

```
Lemma neq_ltVgt (x y : creal) : x ≠ y →{x < y} + {y < x}.
```

where the operator `+` is the disjunction in `Type`.

2.4 Arithmetic operations on Cauchy reals

We build the negation, addition and multiplication on Cauchy reals and prove their output are Cauchy sequences in a systematic way: we perform the appropriate operation on each element of the sequence and we forge a convergence modulus for each operation.

To build the negation, addition and multiplication, we exhibit the convergence moduli of negation, addition and multiplication of Cauchy reals. Given the convergence modulus m_x of $\bar{x}$, we prove the convergence moduli of $(-x_n)_n$, $(x_n + y_n)_n$ and $(x_n y_n)_n$ are respectively: m_x, $\varepsilon \mapsto \max\left(m_x\left(\frac{\varepsilon}{2}\right), m_y\left(\frac{\varepsilon}{2}\right)\right)$ and $\varepsilon \mapsto \max\left(m_x\left(\frac{\varepsilon}{2\lceil y\rceil}\right), m_y\left(\frac{\varepsilon}{2\lceil y\rceil}\right)\right)$

To build the inverse, we need to know a non-negative lower bound δ for the sequence $(|x_n|)_n$ of absolute values from some arbitrary rank, and use it to prove that the sequence of point-wise inverses $(\frac{1}{x_n})_n$ is a Cauchy sequence. According to Section 2.2, such a non-negative lower bound δ is given by `(lbound x_neq0)` when given a proof `(x_neq0 : x ≠ 0)` that $\bar{x}$ is apart from 0 (in the sense of Cauchy sequences). This value δ is such that $\forall i > m_x(\delta),\ \delta \leq |xi|$

If i and j are greater than $m_x(\varepsilon\delta^2)$, we have $|x_i - x_j| < \varepsilon\delta^2$ By definition of δ and if i and j are greater than $m_x(\delta)$, we get $\delta \leq |x_i|$ and $\delta \leq |x_j|$, thus $|x_i - x_j| < \varepsilon|x_i x_j|$. And finally:

$$\left|\frac{1}{x_i} - \frac{1}{x_j}\right| < \varepsilon$$

Thus, a convergence modulus is $\varepsilon \mapsto \max\left(m_x(\varepsilon\delta^2), m_x(\delta)\right)$

Morphism property of arithmetic operations. We can check that all arithmetic operations are compatible with the equality for Cauchy sequences, using a simple point-wise study. The order relation is also a compatible. However, there is no need to systematically study the compatibility with apartness.

2.5 Bounds and evaluation for polynomials

Using the Taylor expansion of polynomial P, we define the following bounds:

$$B_0(P, c, r) = 1 + \sum_{i=0}^{n} |p_i|(|c| + |r|)^i$$

$$B_1(P, c, r) = \max(1, 2r)^n \left(1 + \sum_{i=1}^{n} \frac{B_0(P^{(i)}, c, r)}{i!}\right)$$

$$B_2(P, c, r) = \max(1, 2r)^{n-1} \left(1 + \sum_{i=2}^{n} \frac{B_0(P^{(i)}, c, r)}{i!}\right)$$

These bounds satisfy the following properties, for all x and y in $[c - r, c + r]$:

$$|P(x)| \leq B_0(P, c, r)$$
$$|P(y) - P(x)| \leq |y - x| B_1(P, c, r)$$
$$\left|\frac{P(y) - P(x)}{y - x} - P'(x)\right| \leq |y - x| B_2(P, c, r)$$

These bounds are constructive witnesses for well-known classical mathematical results on continuous or derivable functions, specialized to univariate polynomials. The bound B_0 is only an intermediate step to bounds B_1 and B_2. The bound B_2 is used in Section 4.2 to prove that polynomials whose derivative does not change sign on an interval are monotone on it.

The bound B_1 is used to show that polynomial evaluation preserves the Cauchy property for sequences. Indeed, we build polynomial evaluation of a polynomial $P \in F[X]$ in a Cauchy real as the point-wise operation, and in order to prove that the result is a Cauchy sequence, we bound $|P(x) - P(y)|$ when $|x - y|$ is small enough. The convergence modulus is given by $\varepsilon \mapsto m_x\left(\frac{\varepsilon}{B_1(P, 0, \lceil x \rceil)}\right)$. We then prove that $P(\bar{x}) \neq P(\bar{y}) \Rightarrow \bar{x} \neq \bar{y}$, which implies that $\bar{x} \equiv \bar{y} \Rightarrow P(\bar{x}) \equiv P(\bar{y})$, hence the evaluation of a polynomial in a Cauchy real is compatible with the equality of Cauchy reals.

3 An existential type for algebraic Cauchy reals

3.1 Construction of algebraic Cauchy reals

Now, we formalize real algebraic numbers on top of Cauchy reals.

```
Inductive algcreal := AlgCReal {
  creal_of_alg : creal;
  annul_algcreal : {poly F};
  _ : monic annul_algcreal;
  _ : annul_algcreal.[creal_of_alg] ≡ 0
}.
```

Here, an algebraic Cauchy real `(AlgCReal x P monic_P root_P_x)` represents an algebraic number as a Cauchy real `x` and a polynomial `P` with a proof `monic_P` that `P` is monic (its leading coefficient is 1) and a proof `root_P_x` that `x` is a root of `P`. The notation `p.[x]` stands for polynomial evaluation in the source code.

First we prove that Cauchy reals setoid equality is decidable on algebraic Cauchy reals, then we build arithmetic operations.

3.2 Equality decision procedure

Whereas the comparison on Cauchy reals is only semi-decidable, the comparison on algebraic Cauchy reals is decidable. We call `eq_algcreal` this decision procedure. It uses the additional data given by the annihilating polynomials. In fact, we only need to decide if some algebraic Cauchy real is zero, because we can test whether $\bar{x} = \bar{y}$ by comparing $\bar{x} - \bar{y}$ to 0 once we have the subtraction.

Let $(\bar{x}, P)$ be an algebraic Cauchy real we wish to compare to 0, so P is the annihilating polynomial of the Cauchy real $\bar{x}$. There are two possibilities:

- *Either the indeterminate X does not divide P*, then 0 is not a root of P, thus $\bar{x} \neq 0$.
- *Or X divides P*. If $P = X$ then $\bar{x} \equiv 0$, so let us suppose that X is a proper divisor of P. Then there exist a divisor D of P whose degree is smaller than the one of P and such that $D(\bar{x}) \equiv 0$. The existence of such a D is given by a general lemma stating that if $\bar{x}$ is a Cauchy real and P, Q two polynomials that are not coprime and such that $P(\bar{x}) \equiv 0$ and P does not divide Q, then there exist D of smaller degree than P such that $D(\bar{x}) \equiv 0$.
  We can now iterate this reasoning on $(\bar{x}, D)$ where the degree of D is smaller than the one of P.

3.3 Operations on algebraic Cauchy reals

We build all the operations (negation, addition, multiplication, inverse) from the constants 0 and 1 and using the subtraction and the division. The embedding of the constants $c \in F$ is obtained from the pair $(\bar{c}, X - c)$ (where $\bar{c}$ is a constant Cauchy sequence).

In the remainder of this section we consider two algebraic Cauchy reals x and y, whose respective Cauchy sequences are $\bar{x}$ and $\bar{y}$, and whose respective annihilating polynomials are P and Q.

Let us recall (Section 2.4) that the subtraction $\bar{x} - \bar{y}$ (resp. division $\frac{\bar{x}}{\bar{y}}$) is obtained as the point-wise subtraction (resp. division) of elements of the sequence. Let us find a polynomial whose root is this new sequence.

Subtraction Our candidate is the following resultant:

$$R(Y) = \mathrm{Res}_X\left(P(X+Y), Q(X)\right)$$

There are two essential properties to prove about this resultant it is non zero and it annihilates the subtraction.

R is non zero. Let us suppose that R is zero and find a contradiction. Since R is zero, $P(X+Y)$ and $Q(X)$ are not coprime.

Thanks to the corollary to Bézout theorem, we know there exist $U, V \in F[X]$ such that U is non zero, $\deg_X(U) < \deg(Q)$ and $U(X,Y)P(X+Y) = V(X,Y)Q(X)$.

Taking the Y-leading coefficient, we get $u(X)p = v(X)Q(X)$ where $u(X)$ and $v(X)$ are the respective Y-leading coefficients of $U(X,Y)$ and $V(X,Y)$, and p is the leading coefficient of P. This equation gives that $\deg(Q) \leq \deg(u)$, but $\deg(u) \leq \deg_X(U) < \deg(Q)$. This is a contradiction.

R annihilates the subtraction. Let us prove that R annihilates the Cauchy sequence $\bar{x} - \bar{y}$. Since R is in the ideal generated by $P(X+Y)$ and $Q(X)$, there exist U and V such that $R(Y) = U(X,Y)P(X+Y) + V(X,Y)Q(X)$. Hence by evaluation at $X = y_n$ and $Y = (x_n - y_n)$:

$$R(x_n - y_n) = U(y_n, x_n - y_n)P(x_n) + V(y_n, x_n - y_n)Q(y_n)$$

But $P(\bar{x}) \equiv 0$ and $Q(\bar{y}) \equiv 0$. As x_n and y_n are bounded and U is bounded on a bounded domain (cf Section 2.5) we have that $R(\bar{x} - \bar{y}) \equiv 0$.

Remark that now the subtraction is defined, we can decide the equality of two arbitrary values by comparing their subtraction to zero, using the result from Section 3.2.

Division When $\bar{y}$ is zero, we return the annihilating polynomial X. When it is non zero, we can find a new Q annihilating $\bar{y}$ such that $Q(0) \neq 0$. The annihilating polynomial of $\frac{\bar{x}}{\bar{y}}$ is the following resultant:

$$R(Y) = \mathrm{Res}_X\left(P(XY), Q(X)\right)$$

R is non zero. Let us suppose that R is zero and find a contradiction. Since R is zero, $P(XY)$ and $Q(X)$ are not coprime.

Thanks to the corollary to Bézout theorem, we know there exist $U, V \in F[X]$ such that U is non zero, $\deg_X(U) < \deg(Q)$ and $U(X,Y)P(XY) = V(X,Y)Q(X)$.

By evaluation at $Y = 0$ we get: $U(X,0)P(0) = V(X,0)Q(X)$. Since $F[Y]$ is an integral domain, if $V(X,0) = 0$ we know that $Y|V(X,Y)$, and that there are two possibilities:

- Either $U(X,0) = 0$, which means $Y|U(X,Y)$. Hence, there exists $U'(X,Y)$ and $V'(X,Y)$, whose degrees in Y are strictly smaller than the ones of U and V, and such that: $U'(X,Y)P(XY) = V'(X,Y)Q(X)$.
- Or $P(0) = 0$, which means $X|P(X)$, thus $XY|P(XY)$. But we also know that $U(0,Y)P(0) = V(0,Y)Q(0)$. And since $Q(0) \neq 0$, we necessarily have $V(0,Y) = 0$. It follows that $X|V(X,Y)$ and as we knew that $Y|V(X,Y)$, we find that $XY|V(X,Y)$.
  Thus, there exist P' and V' whose degrees are strictly smaller than those of P and V respectively, such that $U(X,Y)P'(XY) = V'(X,Y)Q(X)$.

In both cases, we can repeat the same reasoning until we get an equation of the following form, such that no member cancels: $U(X,0)P(0) = V(X,0)Q(X)$. This equation gives $\deg(Q) \leq \deg(U(X,0))$, but we also had $\deg(U(X,0)) \leq \deg_X(U) < \deg(Q)$. This is a contradiction.

R annihilates the division. In the same way we did for subtraction, we show that $R(\frac{\bar{x}}{\bar{y}}) \equiv 0$.

4 Encoding algebraic Cauchy reals

The data-type of algebraic Cauchy reals is a setoid whose equivalence is decidable, and it is difficult to show that algebraic Cauchy reals form a countable setoid if F is countable. However, we can do better and build a type whose decidable equivalence reflects Leibniz equality, and for which we can exhibit a bijection with $\mathbb{N}$ if F is countable.

In order to get the type of real algebraic numbers, we should quotient the type of algebraic Cauchy reals by the setoid equality. We know from [3] that this quotient can be done inside Coq as soon as the type which gets quotiented has a `choiceType` structure and the equivalence relation by which we quotient is decidable. Since `algcreal` cannot directly be equipped with a `choiceType` structure, we create a type `algdom` which we call real algebraic domain. The type `algdom` only serves as an encoding of `algcreal` in order to forge the quotient, the construction of which we detail in Section 5.

```
Inductive algdom := AlgRealDom {
  annul_algdom : {poly F};
  center_alg : F;
  radius_alg : F;
  _ : monic annul_algdom;
  _ : annul_algdom.[center_alg - radius_alg]
      * annul_algdom.[center_alg + radius_alg] ≤ 0
}.
```

An element `(AlgRealDom P c r monic_P chg_sign_P)` of `algdom` represents one of the roots of the polynomial `P` in the interval `[c - r, c + r]`, with a proof `monic_P` that `P` is monic and a proof `chg_sign_P` that `P` changes sign on the interval. We know which root is selected by running the decoding procedure described in Section 4.1.

This data-type is only using elements of F and two proofs. It thus can be encoded as sequences of elements of F and inherits the `choiceType` structure of F. We also notice that `algdom` is countable as soon as F is. This fact was not obvious for the setoid of algebraic Cauchy reals. The quotient type will also inherit from the `choiceType` structure and will be countable if F is.

We show that `algdom` is an explicit encoding of algebraic Cauchy reals. Remark that `algcreal` is still useful because arithmetic operations are easier to define on it.

4.1 Decoding to algebraic Cauchy reals

We build the decoding function `to_algcreal : algdom → algcreal`.

An element from the real algebraic domain contains a polynomial P, a center c and a radius r such that $P(c-r)P(c+r) \leq 0$. The root we wish to select is in the interval $I = [c-r, c+r]$.

We decode an element from the real algebraic domain into an algebraic Cauchy real by dichotomy. We form the Cauchy sequence $\bar{x} = (x_n)_n$, such that all the x_n are in the interval I and such that $P(\bar{x}) \equiv \bar{0}$.

We proceed by induction on n to define the sequence $\bar{x}$. It should satisfy the following invariant, which expresses that P must change sign on the interval of radius $2^{-n}r$ and centered in x_n:

$$H_n = P(x_n - 2^{-n}r)P(x_n + 2^{-n}r) \leq 0$$

In the induction step, we pick either $x_n - 2^{-(n+1)}r$ or $x_n + 2^{-(n+1)}r$ to satisfy the invariant H_{n+1}.

The condition that it changes sign is sufficient to show the existence of a root, and doesn't assert anything about its unicity. However, we have no need for unicity as the decoding procedure selects a root in a deterministic manner.

4.2 Encoding of algebraic Cauchy reals

This step is more difficult, we construct the encoding function `to_algdom : algcreal → algdom`. In order to satisfy the coding property:

```
Lemma to_algdomK x : to_algcreal (to_algdom x) ≡ x.
```

Given an algebraic Cauchy real $(\bar{x}, P)$ we try to find a rational interval containing only one root, in order for the decoding to return an element equivalent to $\bar{x}$.

There are two possibilities:

- *Either P and its derivative P' are coprime,* so there exist U and V such that $UP + VP' = 1$. Since $P(\bar{x})$ converges to 0 and if n is big enough we get $P'(x_n) \geq \frac{1}{2\lceil V(\bar{x})\rceil}$. By taking a small enough interval $[a, b]$ containing x_n, we get that P is monotone on $[a, b]$ (thanks to the B_2 bound of Section 2.5) Without loss of generality, we can suppose that P is increasing, we then get $P(a) \leq P(x_i) \leq P(b)$ for all $i \geq n$. But $P(x_i)$ converges to 0, so $P(a) \leq 0 \leq P(b)$. We found an interval with only one root for P.
- *Or P and P' are not coprime,* so we can find a proper divisor D of P that still annihilates x, thanks to the same general lemma mentioned in Section 3.2, in the second case of the disjunction. We fall back to the study of $(\bar{x}, D)$, where the degree of D is strictly smaller that the one of P.

4.3 Transferring the operations to the encoding

We can transpose all the operations and properties of algebraic Cauchy reals to its encoding real algebraic domain. More particularly, equality between algebraic Cauchy reals $\equiv$ (which we showed decidable in 3.2) gives a decidable equivalence on real algebraic domain, using the following definition:

```
eq_algdom x y := (eq_algcreal (to_algcreal x) (to_algcreal y))
```

All the properties of these new operators are easily derived from the properties of the original operators.

5 Real algebraic numbers as a quotient type

The construction of the quotient is done in a generic way, but for this paper to be self-contained, we describe its construction as it is automatically done by the mechanism presented in [3].

5.1 Construction of the quotient type

First we define a notion of canonical element. To each element `x` in `algdom`, we associate an element `(canon x)` which must be equal to any `(canon y)` if and only if `eq_algdom x y`. We use the unique choice operator `xchoose` to do this:

```
Lemma exists_eq (y : algdom) : ∃x : algdom, y ≡x.
Proof. exists y; reflexivity. Qed.

Definition canon (y : algdom) = xchoose (exists_eq y).
```

Moreover, `canon` is constant on each equivalence class thanks to the unicity property of `xchoose`.

Then we define the quotient type of real algebraic numbers by forming the sigma-type of elements of the real algebraic domain that are canonical:

```
Definition alg := {x : algdom | canon x = x}
```

Thanks to the uniqueness of equality proofs on `algdom`, two elements `x` and `y` in `alg` are equal if and only if `(val x = val y)`, where `val` is the projection on the first component of the sigma-type. From `canon`, we can now build the canonical surjection `(pi : algdom →alg)`, which maps any element of `algdom` to the unique representative for its equivalence class.

By composing `to_algdom` with `pi` we can now see `alg` as the type of equivalence classes of elements of `algcreal`. We now see `F` as a parameter for the whole construction, so that `alg` becomes `(alg F)`, which we denote by $\bar{F}$.

We prove that arithmetic operations (and the order relation) are compatible with the quotient. This is a direct consequence of the morphism property of operations with regard to setoid equality, which we dealt with in Section 3.3.

We also build a function `(to_alg: F →alg F)` which embeds any element c of F into $\bar{F}$, by mapping c to the equivalence class of the element $(\bar{c}, (X - c))$

of `algcreal`. We then prove it is a field morphism and that this morphism is also compatible with comparison. The mathematical notation for this function is $\uparrow$.

We remark that by construction of `algdom`, the following property holds: given a polynomial $P \in F[X]$ and two points $a < b \in F$ such that $P(a) \leq 0 \leq P(b)$, there exist $c \in \bar{F}$ such that $c \in [a, b]$ and $P(c) = 0$. This is a weak version of the intermediate value property for polynomials.

5.2 Real algebraic numbers form a real closed field

Note that $\bar{F}$ is a totally ordered Archimedean field with decidable comparison. Indeed, as those properties already hold for F, they transfer to $\bar{F}$ by studying the Cauchy sequences underlying its elements.

The difficulty is to prove $\bar{F}$ is a real closed field, which amounts to prove the intermediate value theorem for polynomials in $\bar{F}[X]$.

Let P be a polynomial in $\bar{F}[X]$ and a and b two elements of $\bar{F}$ such that $a < b$ and $P(a) \leq 0 \leq P(b)$. Let us show that there exist an real algebraic number c in $\bar{F}$ such that $P(c) = 0$.

Iteration of the closure. Thanks to the remark in the end of Section 5.1, applied to the ordered Archimedean field $\bar{F}$, we get that the polynomial $P \in \bar{F}[X]$ has a root ξ in the "double closure" $\bar{\bar{F}}$.

If we find a function $\downarrow\colon \bar{\bar{F}} \to \bar{F}$, such that $\forall \zeta \in \bar{\bar{F}}, \quad \uparrow (\downarrow \zeta) = \zeta$, then $(\downarrow \xi) \in \bar{F}$ would be a root of P. The Coq name for this function is `from_alg`. The existence of such a function means that the closure process we design terminates in one step only.

Let ξ be in $\bar{\bar{F}}$, and let us build $(\downarrow \xi)$. By transforming ξ in an algebraic Cauchy real $(\bar{\xi}, P)$ we get a Cauchy sequence $\bar{\xi}$ in $\bar{F}^{\mathbb{N}}$, and a polynomial $P \in \bar{F}[X]$.

Each element ξ_n is a Cauchy sequence $\bar{x}_n = (x_{n,k})_k$ which we can choose such that $|\bar{x}_{n+1} - \bar{x}_n| < 2^{-(n+1)}$. Then, the sequence $\bar{x} = (x_{n,n})_n$ is a Cauchy sequence such that $\uparrow \bar{x} = \bar{\xi}$. We hence have the first component of $(\downarrow \xi)$.

Polynomial annihilating the algebraic Cauchy real $\bar{x}$. We must find a polynomial $R \in F[X]$ which annihilates $\bar{x}$. The coefficients p_i of P are a finite number of values in the field extension $\bar{F}$ of F, so we can apply the primitive element theorem to find an element $\alpha \in \bar{F}$, whose annihilating polynomial is Q of degree $q + 1$ such that for all i, p_i is in the simple extension $F[\alpha]$. We can then re-factorize P as $P = \sum_{l=0}^{q} \alpha^l P_l$.

We take the resultant $R(Y) = \mathrm{Res}_X \left(\sum_{l=0}^{q} X^l P_l(Y), Q(X) \right)$. We now show that it is non zero and it annihilates $\bar{x}$.

R is non zero. Let us suppose R is zero and find a contradiction. The property of Bézout gives $U, V \in F[X]$ such that U is non zero, $\deg_X(U) < \deg(Q)$ and:

$$U(X, Y) \sum_{l=0}^{q} X^l P_l(Y) = V(X, Y) Q(X)$$

Then by embedding in $\bar{F}$ and evaluation at $X = \alpha$ we get: $U(\alpha, Y)P(Y) = 0$. But $P \neq 0$, thus $U(\alpha, Y) = 0$. Then by taking the Y-leading coefficient $u(X)$ of $U(X, Y)$ we get:

$$u(\alpha) = 0 \quad \text{and} \quad u \in F[X] \quad \text{and} \quad u \neq 0 \quad \text{and} \quad \deg(u) < \deg(Q)$$

This gives a polynomial u annihilating α of degree smaller than the one of Q, and we can proceed by induction on the degree of Q.

R annihilates $\bar{x}$. We have:

$$R(x_{n,n}) = U(\alpha_m, x_{n,n}) \left(\sum_{l=0}^{q} \alpha_m^l P_l(x_{n,n}) \right) + V(\alpha_m, x_{n,n}) Q(\alpha_m)$$

and we notice that the right hand side converges to 0 when m and n grow.

Conclusion

The theory of real closed fields presented in [5] is based on an interface we now provide an instance of. A direct consequence is that real algebraic numbers immediately enjoy quantifier elimination which proves decidable its first order theory. The formalization we describe comes from various classical sources that had to be adapted, made constructive and simplified for the needs of the formalization. The methodology applied here to build algebraic numbers and make proofs feasible and quick is, up to our knowledge, original. This is also, as far as we know, the first certified formalization of real algebraic numbers in a proof assistant.

It would be interesting to provide an efficient implementation of algebraic numbers, relying on [2] and on [9] for example. The formalization we show in this paper would then serve as a reference implementation. We would need to prove the relative correctness of the efficient implementation with regard to the abstract one. But no proofs about the algebraic structure of the new implementation would be required.

It would be natural to continue this work by extending the real algebraic numbers by the imaginary unit $\mathbf{i}$. Thanks to the constructive fundamental algebra theorem, generalized to real closed fields [4], this new field would be *algebraically closed*, *partially ordered* and would then represent the data-type of *(complex) algebraic numbers*. In the framework of Galois theory, it would also be interesting to formalize the type of algebraic extensions over rational numbers: we could then use the classical presentation and study them into their algebraic closure.

Finally, we formalized the construction of the real closure of fields of zero characteristic, which is a step in constructing the algebraic closure. It is a completely different work to formalize the algebraic closure of fields of non-zero characteristic. Moreover the efficient algorithms for the non-zero characteristic are treated in [2] and are more intricate than the ones for the zero characteristic.

Acknowledgement

I wish to thank Georges Gonthier for the numerous ideas which constitute the basis of this development and Russell O'Connor for discussions which helped me find the good way to state and prove some results. I also thank Assia Mahboubi, Enrico Tassi and the anonymous referees for their reading and comments on this paper.

References

1. Barthe, G., Capretta, V., Pons, O.: Setoids in type theory. J. of Functional Programming 13(2), 261–293 (2003), Special Issue on Logical Frameworks and Metalanguages
2. Bostan, A.: Algorithmique efficace pour des opérations de base en Calcul formel. Ph.D. thesis, École polytechnique (2003), http://algo.inria.fr/bostan/these/These.pdf
3. Cohen, C.: Types quotients en Coq. In: Hermann (ed.) Actes des 21ème journées francophones des langages applicatifs (JFLA 2010). INRIA, Vieux-Port La Ciotat, France (Jan 2010), http://jfla.inria.fr/2010/actes/PDF/cyrilcohen.pdf
4. Cohen, C., Coquand, T.: A constructive version of Laplace's proof on the existence of complex roots, http://hal.inria.fr/inria-00592284/PDF/laplace.pdf, unpublished
5. Cohen, C., Mahboubi, A.: Formal proofs in real algebraic geometry: from ordered fields to quantifier elimination. Logical Methods in Computer Science 8(1:02), 1–40 (Feb 2012), http://hal.inria.fr/inria-00593738
6. Delahaye, D.: A Tactic Language for the System Coq. In: Parigot, M., Voronkov, A. (eds.) Logic for Programming and Automated Reasoning (LPAR). Lecture Notes in Computer Science (LNCS)/Lecture Notes in Artificial Intelligence (LNAI), vol. 1955, pp. 85–95. Springer, Reunion Island (France) (Nov 2000)
7. Garillot, F., Gonthier, G., Mahboubi, A., Rideau, L.: Packaging mathematical structures. In: Berghofer, S., Nipkow, T., Urban, C., Wenzel, M. (eds.) TPHOLs. Lecture Notes in Computer Science, vol. 5674, pp. 327–342. Springer (2009)
8. Geuvers, H., Niqui, M.: Constructive reals in coq: Axioms and categoricity. In: Selected papers from the International Workshop on Types for Proofs and Programs. pp. 79–95. TYPES '00, Springer-Verlag, London, UK (2002), http://dl.acm.org/citation.cfm?id=646540.696040
9. Krebbers, R., Spitters, B.: Computer certified efficient exact reals in coq. In: Conference on Intelligent Computer Mathematics, CICM 2011 Proceedings. Lecture Notes in Artifical Intelligence, Springer (2011)
10. Lang, S.: Algebra. Graduate texts in mathematics, Springer (2002)
11. Mines, R., Richman, F., Ruitenburg, W.: A course in constructive algebra. Universitext (1979), Springer-Verlag (1988)
12. O'Connor, R.: Certified exact transcendental real number computation in coq. In: Mohamed, O., Muñoz, C., Tahar, S. (eds.) Theorem Proving in Higher Order Logics, Lecture Notes in Computer Science, vol. 5170, pp. 246–261. Springer Berlin / Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-71067-7_21
13. Project, T.M.C.: SSReflect extension and libraries. http://www.msr-inria.inria.fr/Projects/math-components/index_html