

# Construction of Real Algebraic Numbers in Coq

**Cyril Cohen**

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

August 13, 2012

# Why algebraic numbers ?

- Field strictly between  $\mathbb{Q}$  and  $\mathbb{R}$
- Instance of real closed field (RCF), countable and with decidable comparison
- Useful in the formalization of the odd order theorem (Mathematical Components project)
- Interesting object in real algebraic geometry and computer algebra

# Usual definition of algebraic reals

Algebraic numbers are roots of non null polynomials  
with coefficients in  $\mathbb{Q}$

Classically:

$$\{x \in \mathbb{R} \mid \exists P \in \mathbb{Q}[X] - \{0\}, P(x) = 0\}$$

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# “Top-down” approach

- We start from real numbers
- We restrict them to roots of polynomials with coefficients in  $\mathbb{Q}$

# Cauchy reals

Representation of a Cauchy real  $\bar{x}$

- A sequence  $(x_n)_{n \in \mathbb{N}}$
- A convergence modulus  $m_x : \mathbb{Q} \rightarrow \mathbb{N}$   
such that :  $\forall \varepsilon \ i \ j, m_x(\varepsilon) < i, j \rightarrow |x_i - x_j| < \varepsilon$

# The theory of Cauchy reals

- A definition of  $\equiv$  (non decidable)

$$\bar{x} \equiv \bar{y} \iff |x_n - y_n| \xrightarrow[n \rightarrow \infty]{} 0$$

- Cauchy reals form a setoid
- Arithmetic operations are morphisms
- Properties of ordered fields hold

# Apartness versus Equality

- The primitive notion is apartness:  $\bar{x} \neq \bar{y}$
- $x \equiv y$  is defined by  $\neg \bar{x} \neq \bar{y}$
- It is not true that  $\neg \bar{x} \equiv \bar{y} \Rightarrow \bar{x} \neq \bar{y}$

Apartness is more informative than equality.



# Sum of two Cauchy reals

The sum of  $\bar{x}$  and  $\bar{y}$  is given by:

- the sequence  $(x_i + y_i)_i$
- the convergence modulus  
 $\varepsilon \mapsto \max \left( m_x \left( \frac{\varepsilon}{2} \right), m_y \left( \frac{\varepsilon}{2} \right) \right)$

# Reciprocal of a Cauchy real

To produce the reciprocal of  $\bar{x}$ , we need to bound

$$\left| \frac{1}{x_n} - \frac{1}{x_m} \right|$$

with some Cauchy modulus.

$$\left| \frac{1}{x_n} - \frac{1}{x_m} \right| = \left| \frac{x_m - x_n}{x_n x_m} \right|$$

$\Rightarrow$  It suffices to bound  $x_n$  below.

$\Rightarrow$  It suffices to have a proof of  $\bar{x} \neq 0$

inv\_creal :

`forall x : creal -> (x != 0) -> creal`

# “Top-down” representation

We define the type algcreal of algebraic Cauchy reals:

- A Cauchy real  $\bar{x}$
- A monic polynomial  $P \in \mathbb{Q}[X]$  such that  $P(\bar{x}) \equiv 0$

# Properties of algebraic Cauchy reals

- Comparison becomes decidable (thanks to the polynomial)
  - We translate all the operations from Cauchy reals
- We need to rebuild a new polynomial for each operation (thanks to the computation of a resultant).

# Computation of the subtraction

Let  $x = (\bar{x}, P)$  and  $y = (\bar{y}, Q)$  two algebraic Cauchy reals.

The subtraction  $x - y$  is given by  $(\bar{x} - \bar{y}, R)$  with

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

# Comparison

Lemma neq\_ltVgt (x y : creal) : x != y ->  
{x < y} + {y < x}.

# Equality test

$x \equiv y$  is equivalent to  $x - y \equiv 0$

$\Rightarrow$  We study  $(\bar{x}, P) \equiv 0$

# Equality test to 0

We study  $(\bar{x}, P) \equiv 0$  by induction on the degree of  $P$ .

- either  $X \nmid P$  then  $P(\bar{x}) \neq 0$  and  $\bar{x} \neq 0$
- or there exists  $Q$  such that  $P = QX$  and  $Q(\bar{x}) \equiv 0$  or  $X(\bar{x}) \equiv 0$ 
  - in the first case, we recur using  $Q$
  - in the second case, we have exactly  $\bar{x} \equiv 0$



# Key lemma

**Lemma** `poly_mul_creal_eq0`  $P \ Q \ x :$   
 $P.[x] * Q.[x] == 0 \rightarrow$   
 $\{P.[x] == 0\} + \{Q.[x] == 0\}.$

Ideas :

- reduce to  $P$  and  $Q$  coprime
- there exists polynomials  $U$  and  $V$ , such that  $UP + VQ = 1$ .
- for big enough values of  $n$ ,  
$$U(x_n)P(x_n) + V(x_n)Q(x_n) > \frac{1}{2}$$

# Reciprocal of an algebraic Cauchy real

Reciprocal is now a total unary operator `algcreal`.  
(No apartness proof is required anymore)

Definition `inv_algcreal`

```
(x : algcreal) : algcreal :=
match eq_algcreal_dec x (cst_algcreal 0)
with
| left x_eq0 => cst_algcreal 0
| right x_neq0 => @AlgCReal
  (@inv_creal _ x_neq0) _
  (@monic_annul_creal _)
  (@root_inv_algcreal _ x_neq0)
end.
```

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# “Bottom-up” approach

We represent an algebraic thanks to:

- A polynomial from  $\mathbb{Q}[X] - \{0\}$
- A root selection method (interval, approximation, ...)

# Choice of a selection method

We define the type algdom of the algebraic real domain.

- A monic polynomial  $P \in \mathbb{Q}[X]$
- An interval  $[c - r, c + r]$  such that  $P$  changes sign and is monotone

# Choice of a selection method

We define the type algdom of the algebraic real domain.

- A monic polynomial  $P \in \mathbb{Q}[X]$
- An interval  $[c - r, c + r]$  such that  $P$  changes sign

and built such that  $P$  is monotone

# Encoding algebraic Cauchy reals

We built an encoding and a decoding function:

- to\_algdom: algcreal  $\rightarrow$  algdom
- to\_algcreal: algdom  $\rightarrow$  algcreal

Which mean they satisfy:

`forall` x, to\_algcreal (to\_algdom x) == x



# Decoding

The decoding function proceeds by dichotomy:

Given  $(P, [c - r, c + r])$  we construct the algebraic Cauchy real  $(\bar{x}, P)$ , where  $x_n$  are rationals recursively defined as:

- $x_0 = c$
- $x_{n+1} = x_n - r2^{-(n+1)}$  if  $P$  changes sign on  $[x_n - r2^{-n}, x_n]$
- $x_{n+1} = x_n + r2^{-(n+1)}$  otherwise

# Encoding

Given  $(\bar{x}, P)$  we construct a pair  $(P, [a, b])$  such that:

- $P$  is square free
- $P$  is monotone on  $[a, b]$

# Transfer of the comparison and operators

We trivial deduce operations of `algdom` from those of `algcreal`:

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

```
add_algdom x y := to_algdom (add_algcreal  
  (to_algcreal x)(to_algcreal y))
```

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# Quotient of algdom

`eq_algdom` (again noted `==`) is an decidable equivalence relation on a countable type.

# Quotient of `algdom`

`eq_algdom` (again noted `==`) is an decidable equivalence relation on a countable type.

⇒ We can take the effective quotient of `algdom` by this equivalence.

# Quotient of `algdom`

`eq_algdom` (again noted `==`) is an decidable equivalence relation on a countable type.

- ⇒ We can take the effective quotient of `algdom` by this equivalence.
- ⇒ This gives the type `realalg` of exact real algebraic numbers.



# About the choice

If  $T$  is countable and  $P : T \rightarrow \text{Prop}$  is decidable, there exists a unique-choice function:

xchoose : (`exists`  $x : T, P x$ )  $\rightarrow T$

# “Inlined” construction of `realalg`

Choice of a canonical element :

```
((P y) := (fun x => y == x))
```

```
Lemma exists_eq (y : algdom) :  
  exists x : algdom, y == x.
```

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

Definition of algebraics

```
Definition realalg :=  
  {x : algdom | canon x == x}.
```

# Structure of `realalg`

- We transfer the comparison and the arithmetic operations from `algdom` to `realalg`
  - We already know that they are morphisms for the `algdom` setoid
  - Transferring the equality decision procedure gives a relation that reflects Leibniz equality
- We transfer the properties of operations (deduced from `algcreal`)

# Structure of `realalg`

- We transfer the comparison and the arithmetic operations from `algdom` to `realalg`
    - We already know that they are morphisms for the `algdom` setoid
    - Transferring the equality decision procedure gives a relation that reflects Leibniz equality
  - We transfer the properties of operations (deduced from `algcreal`)
- ⇒ `realalg` has a real closed field structure with decidable comparison

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

# Wishlist

- A Comparison algorithm (equality + order)
- The equality decision procedure reflects Leibniz equality
- The type (or the setoid) is countable
- Arithmetic operations + their properties  
Ordered field
- The intermediate value property for polynomials

- Not appropriate for efficient computation (naive algorithms on naive structures)
- Aims at performing constructive proofs:
  - by providing an implementation to the Real Closed Field structure
  - for proof requiring only reals that are algebraic
  - to help the formalization of an efficient representation

# Conclusion

## Three representations

- A representation to write algorithms :  
algcreal
- A intermediate encoding : algdom
- A “proof-mode” representation : realalg

## Future work :

- Complex algebraics (extension by  $i$ )
- An efficient implementation: a realalg\_eff structure which reflects realalg, and the associated operations



# The End

Thanks for your attention

Questions ?