Constructive axiomatic for the real numbers

Jean-Marie Madiot
Pierre-Marie Pédrot
Junior Laboratory COQTAIL
Ens Lyon - France

May 23, 2011

Reasoning about real numbers in Coq can be very tedious. The standard library of Coq is classical and using it means giving up constructive proofs. C-CoRN is constructive in every way and forbids classical statements. Moreover its hierarchical structure is a bit cumbersome. We propose a short axiomatic that can handle both constructive and classical mathematical statements.

The implementation in Coq is available at COQTAIL's repository[1]

1 Constructive axiomatic

The axiomatic is in form of a module, so that it can be implemented, either with the standard library or with axiom-free Coq terms.

1.1 Ordered ring

The first set of axioms describes the ring operators. $\mathbb{R}$ is a parameter of sort Type, 0 and 1 are two objects of $\mathbb{R}$ and +, *, - are operators on $\mathbb{R}$ respectively of arities 2, 2, 1.

We also need an equality predicate, but first we define the strict order relation and all the other order relations will be defined thanks to this one. Indeed a proof of $x < y$ will contain a computational witness of an ε such that $x + \varepsilon \leq y$. That way, building an inverse of x knowing that $x \neq 0$ will be easier. The sort of the predicate $<$ is not important, as you can use an argument like "constructive epsilon" of the Coq'Art[BC04][2] book to constructively extract such an ε. The axioms on $<$ are asymmetry and transitivity. We then define the relation of discriminability $\#$ as the constructive sum of $<$ and $>$, the inverse relation of $<$. $\equiv$ is the negation of $\#$ and $\leq$ is the constructive sum of $<$ and $\equiv$.

1.2 Constructive field

Now we can define an operator for the inverse, which expects a real number and the proof that this number is discriminable from 0:

$$\cdot^{-1}_{[\cdot]} \ : \ (x : \mathbb{R}), (p : x\#0) \mapsto x^{-1}_{[p]} : \mathbb{R}$$

[1] `https://sourceforge.net/projects/coqtail/develop` in `src/Fresh/Reals`
[2] This trick is also in the standard library of Coq in Logic/ConstructiveEpsilon

We then add the axioms of commutativity, associativity, distributivity of multiplication and/over addition with respect to $\equiv$, and the axioms specifying 0 and 1 as the units for $+$ and $*$. We also specify the inverse operators $-$, $^{-1}$ for $+$, $*$, the latter requiring a proof of $x\#0$.

Moreover we have to say that $+$, $*$, $^{-1}$ and even $\equiv$ behave well with respect to $<$ and $\equiv$:

$$\begin{array}{rcl} x \equiv x' \ , \ x < y & \Rightarrow & x' < y \\ x \equiv x' \ , \ y < x & \Rightarrow & y < x' \\ y \equiv y' & \Rightarrow & x + y \equiv x + y' \\ y < y' & \Rightarrow & x + y < x + y' \\ 0 < x \ , \ y < y' & \Rightarrow & x \cdot y < x \cdot y' \\ 0 < x \ , \ y \equiv y' & \Rightarrow & x \cdot y \equiv x \cdot y' \\ 0 < x \ , \ p' : 0\#x & \Rightarrow & 0 < x^{-1}_{[p']} \end{array}$$

We also require that $0 < 1$ (it is equivalent to $0\#1$, assuming the other axioms).

Then we need the fact that $\mathbb{R}$ is archimedean. This is not constructively equivalent to having powerful tools like the integer part function, as it is not continuous thus not constructively definable. But we can get back from the infinity of $\mathbb{R}$ to finitely representable objects as elements of $\mathbb{Z}$. In fact we can constructively have a function $\llcorner\cdot\lrcorner$ taking the representation of a real number x and returning some integer z such that[3] $|x - \underline{z}| < 1$. We cannot constructively make $\llcorner\cdot\lrcorner$ total and extensional (i.e. $x \equiv y \to \llcorner x \lrcorner = \llcorner y \lrcorner$) because this would make it discontinuous.

Finally we complete $\mathbb{R}$ by adding all the limits of the Cauchy sequences of elements of $\mathbb{R}$.

$$\begin{array}{rcl} \text{cauchy}(u) & := & \forall \varepsilon > 0, \exists N, \forall p, q \geq N \ |u_p - u_q| < \varepsilon \\ u \to l & := & \forall \varepsilon > 0, \exists N, \forall n \geq N, \ |u_n - l| < \varepsilon \end{array}$$

And the axiom we add is:

$$\text{cauchy}(u) \Rightarrow \exists l, \ u \to l$$

With these axioms we hope to capture the constructive real numbers. For instance they do not allow to prove the Markov's principle[4] (semi-constructive [Her10]), the countable principle of omniscience[5] (classical) or the decidability of equality on real numbers (co-semidecidable) all implied by the axioms of the standard library.

2 Constructive/classical distinction

Dealing with real analysis often requires non constructive axioms like the excluded middle or the axiom of choice. However we would like to use extraction of constructive proofs and to know whether the proof of a statement is constructive.

Therefore, if we want to reason constructively we need predicates – like the convergence of a sequence or the differentiability of a function – defined in Type. To reason classically the predicates must be in Prop. A solution to this problem is to duplicate the definitions of the predicates: one for computational content in Type and another for non-constructive statements in Prop. Another is the use of a monad in order to embed constructive predicates into a world where the statements are weaker.

[3] This condition can be replaced by any condition of the form $|x - \underline{\llcorner x \lrcorner}| < 1/2 + \varepsilon$ where $\varepsilon > 0$

[4] Markov's principle: if P is decidable then $\neg\neg(\exists n \ Pn) \Rightarrow \exists n \ Pn$

[5] Countable principle of omniscience: if P is decidable then the property $(\forall n \ Pn)$ is decidable

2.1 Dealing with axioms

Monads allows us to keep track of the external hypothesis used in a proof, through the so-called *axiom* monad

Definition 1. *The axiom monad on X is $TA = X \to A$ with weakening as* `return` *and contraction as* `bind`*.*

Interesting instantiations for X include excluded-middle, choice, epsilon and virtually any axiom independent from pCIC.

This monad makes it possible to have a statically checked analysis of the requirements of a proof. One can also imagine to use a *generic* axiom monad, under the form TXA where X is the list of axioms used.

Remark 1. *In order to make the use of the axiom monads easier, the functional extensionality has to be assumed as most of the time we want to prove properties such as* `lift` $x =$ `lift` y*.*

Sometimes, additional axioms are not that terrible, and one can argue about the burden of the axiom monad. For instance, excluded middle or propositional extensionality are not havoc-wreaking enough to justify embedding them in a monad.

Following Castéran's work [Cas07], non-constructivism can provide an interesting use of the axiom monad. We recall the *epsilon* axiom:

$$\varepsilon \equiv \forall A. \forall (P : A \to \texttt{Prop}).\ (\texttt{exists}\ x, Px) \to \{x \mid P\ x\}$$

The epsilon axiom is problematic because once it has been admitted, one cannot differentiate between proofs that are extractible and those which are not. Indeed, it allows informative content flow between `Prop` and `Type`.

Using the axiom monad on epsilon gives us back the static discrimination between constructive and non-constructive proofs.

2.2 `Prop`-`Type` distinction

A weaker way to encode epsilon is to embed every type in `Prop`. This is done through the inhabited monad.

$$\texttt{Inductive inhabited}\ (\texttt{A} : \texttt{Type}) : \texttt{Prop} := \texttt{inhabits} : \texttt{A} \to \texttt{inhabited A}$$

If we consider using the choice axiom[6], we can recover a equivalent system through the use of epsilon axiom monad.

One needs to take care of this `inhabited` construction. Indeed, whenever proof-irrelevance is assumed, there is at most one proof of `inhabited` A for any A. This implies in particular that the programmer must fully specify the content of the lift through `sig`-like dependent pairs.

For example with the choice axiom, we can specify the partial function inverse on real numbers as follows:

$$\texttt{inhabited}\ \{f\ :\ \mathbb{R} \to \mathbb{R} \mid \forall x\ x\#0 \to fx * x \equiv 1\}$$

[6]The choice axiom, which is purely non-informative:
$\forall (A : \text{Type})\ (B : A \to \text{Type}), (\forall x : A,\ \text{inhabited}(Bx)) \Rightarrow \text{inhabited}(\forall x : A, B\ x)$

References

[BC04] Yves Bertot and Pierre Castéran. Interactive theorem proving and program development. coq'art: The calculus of inductive constructions, 2004.

[Cas07] Pierre Castéran. Utilisation en coq de l'opérateur de description. *Journées Francophones des Langages Applicatifs*, 2007.

[Her10] Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 50–56. IEEE Computer Society, 2010.