Calculemus Autumn School
Pisa September 2002

Formalization in Coq

Formalizing a constructive proof of the Fundamental Theorem of Algebra in Coq

Herman Geuvers, Freek Wiedijk
Jan Zwanenburg, Randy Pollack
Milad Niqui, Henk Barendregt

Fundamental Theorem of Algebra:

Every non-constant polynomial

$$f(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$$

with coefficients in $\mathbb{C}$ has a root
(a $z \in \mathbb{C}$ such that $f(z) = 0$).

- Overview / Motivation
- Constructive proof
- Axiomatic reasoning
- Completely formalized in the theorem prover Coq
- Some details of the proof
- Real numbers

Overview of the FTA project (Motivation)

- Formalize a large piece of real mathematics.
- Library for basic algebra and analysis, to be used by others
- Investigate the current limitations
- Try to manage this project. Three sequential/parallel phases:

Mathematical proof	LaTeX document (lots of details)
Theory development	Coq file (just defs and statements of lemmas)
Proof development	Coq file (proofs filled in)

  Try to keep these phases consistent!

- Constructive proof: reals are (potentially) infinite objects; algorithm.

Constructive proof

- Procedure for (always) finding a root.
- Equality on $\mathbb{R}$ is not decidable.
  (Not: $\forall x, y \in \mathbb{R} \quad (x = y) \vee (x \neq y)$)
- Constructively, a polynomial

$$f(z) = a_n z^n + a_{n-1} z^{n-1} + \ldots + a_1 z + a_0$$

  is non-constant if

  $a_k \# 0$ for some $k > 0$.

Axiomatic reasoning

- First define an algebraic hierarchy
  (semi-groups, monoids, groups, rings, fields, ordered fields)
- Advantages: Reuse of proven results; reuse of notation
- $\mathbb{R}$ is a Cauchy-complete Archimedean ordered field.
- $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- A concrete instantiation for $\mathbb{R}$ has been constructed (Niqui)

Completely formalized in the theorem prover Coq

Sets and Basics	41 kb
Algebra (upto Ordered Fields)	165 kb
Reals	52 kb
Polynomials	113 kb
Real-valued functions / Basic Analysis	30 kb
Complex numbers	98 kb
FTA proof	70 kb
Construction of $\mathbb{R}$ (Niqui)	309 kb
Rational Tactic	49 kb

This is the size of our input files (definitions, lemmas, tactic scripts)

Some details of the proof

- Which proof to formalize?
- There are various constructive proofs of FTA:
  Weyl; Brouwer-de Loor; Bishop; Kneser (also in Troelstra and van Dalen).
- We have followed the Kneser proof.
- It can be seen as a constructive version of 'the' classical FTA proof.

The classical FTA proof

Suppose $|f(z)|$ is minimal with $|f(z)| \neq 0$.
We construct a z_0 with $|f(z_0)| < |f(z)|$.
We may assume that the minimum is reached for $z = 0$.

$$f(x) = a_0 + a_k x^k + O(x^{k+1})$$

with a_k the first coefficient that's not 0.
Now take

$$z_0 := \epsilon \sqrt[k]{-\frac{a_0}{a_k}}$$

with $\epsilon \in \mathbb{R}_{>0}$.
If ϵ is small enough, the part $O\left(z_0^{k+1}\right)$ will be negligible and we get a $z_0 \neq 0$ for which

$$|f(z_0)| = a_0 + a_k \left(\epsilon \sqrt[k]{-\frac{a_0}{a_k}}\right)^k = a_0(1 - \epsilon^k) < |f(0)|$$

The constructive FTA proof

Define an algorithm

Given $z \in \mathbb{C}$, construct a sequence $z, z_0, z_1, \ldots$ going to the root.

Problem: in the definition

$$z_0 := \epsilon \sqrt[k]{-\frac{a_0}{a_k}}$$

- ϵ must be small enough to neglect $O\left(z_0^{k+1}\right)$
- ϵ must be large enough to reach the root.

Solution: write

$$f(x) = a_0 + a_k x^k + \text{other terms}$$

and find k and z_0 such that $|a_k||z_0|^k$ is big enough w.r.t. the other terms and small enough compared to $|a_0|$.

Real Numbers in Coq:

- Axiomatic: a 'Real Number Structure' is a
  Cauchy-complete Archimedean ordered field.
- Prove FTA 'for all real numbers structures'.
- Construct a model to show that real number structures exist.
  (Cauchy sequences over an Arch. ordered field, say $\mathbb{Q}$)
- Prove that any two real number structures are isomorphic.

Axioms for Real Numbers:

- Algebraic hierarchy based on Constructive Setoids.
- Apartness $\#$ as basic

$$x = y \leftrightarrow \neg(x \# y)$$
$$x \# y \rightarrow (x \# z) \vee (y \# z)$$
$$x \# x$$
$$x \# y \rightarrow y \# x$$

- Field operation reciprocal

$$1/- : (\Sigma x{:}F.x \# 0) \rightarrow (\Sigma x{:}F.x \# 0)$$

- Cauchy sequences over Field F:
  g : nat $\rightarrow F$ is Cauchy if

$$\forall \epsilon{:}F_{>0}.\exists N{:}\mathbb{N}.\forall m \geq N(|g_m - g_N| < \epsilon)$$

The algebraic hierarchy of the FTA project

- In proving FTA, we have to deal with real numbers, complex numbers and polynomials
- Many of the properties we use are generic and algebraic.
- To be able to reuse results (also for future developments) we have defined a hierarchy of algebraic structures.
- Basic level: constructive setoids, $\langle A, \#_A, =_A \rangle$, with A : Set, $\#_A$ an apartness and $=_A$ an equivalence relation.
- Next level: semi-groups, $\langle S, + \rangle$, with S a setoid and $+$ an associative binary operation on S.

Inheritance via Coercions

We have the following coercions.

```
OrdField >-> Field >-> Ring >-> Group
                Group >-> Monoid >-> Semi_grp >-> Setoid
```

- All properties of groups are inherited by rings, fields, etc.
- Also notation is inherited:

  `x[+]y`

  denotes the addition of `x` and `y` for `x,y:G` from any semigroup (or monoid, group, ring,...) `G`.
- The coercions must form a tree, so there is no real multiple inheritance:
  E.g. it is not possible to define rings in such a way that it inherits both from its additive group and its multiplicative monoid.

Proofs by computation / Reflection

Needed for the proof of FTA:

Proofs of equalities between rational expressions like

$$\frac{1}{x+y} + \frac{1}{x-y} = \frac{2x}{x^2-y^2}$$

are obtained by partial reflection.

Following the reflection method:

$$[\![-]\!] : E \to \mathbb{R}$$

with E the type of rational expressions. So E contains a constructor `erecip : E -> E`

But in the case of rational expressions, the $[\![-]\!]$ can not be total.

$\rightsquigarrow$ partial reflection.

Axioms for Real Numbers ctd.:

- All Cauchy sequences have a limit:

$$\mathsf{SeqLim} : (\Sigma g{:}\mathsf{nat}{\to}F.\mathsf{Cauchy}\, g) \to F$$

$$\mathsf{CauchyProp} : \forall g{:}\mathsf{nat}{\to}F.(\mathsf{Cauchy}\, g) \to \forall \epsilon{:}F_{>0}.\exists N{:}\mathbb{N}.\forall m \geq N.(|g_m - (\mathsf{SeqLim}\, g)| < \epsilon)$$

- Axiom of Archimedes: (there are no non-standard elements)

$$\forall x{:}F.\exists n{:}\mathbb{N}(n > x)$$

NB: The axiom of Archimedes proves that 'ϵ-Cauchy sequences' and '$\frac{1}{k}$-Cauchy sequences' coincide (similar for limits):
Viz: $g : \mathsf{nat} \to F$ is a $\frac{1}{k}$-Cauchy sequence if

$$\forall k{:}\mathbb{N}.\exists N{:}\mathbb{N}.\forall m \geq N(|g_m - g_N| < \frac{1}{k+1})$$

Construction of a Real Number Structure:

- Construct $\mathbb{Q} := \{\langle p, n\rangle \mid p{:}\mathbb{Z}, n{:}\mathbb{N}\}$ with intended interpretation $\langle p, n\rangle \mapsto \frac{p}{n+1}$.
- Define all operations and relations, turning $\mathbb{Q}$ into an Archimedean constructive ordered field.
- For F an (Archimedean) constructive ordered field, define the Cauchy completion of F, CSeq_F.

$$h < g := \exists \epsilon : F_{>0}.\exists N{:}\mathbb{N}\forall m \geq N.(g_m - h_m > \epsilon).$$

$$g_m^{-1} := \begin{cases} 0 & \text{if } \; m < N \\ \frac{1}{g_m} & \text{if } \; m \geq N \end{cases}$$

where N is such that $\forall m \geq N.|g_m| > \epsilon$ for some $\epsilon > 0$.

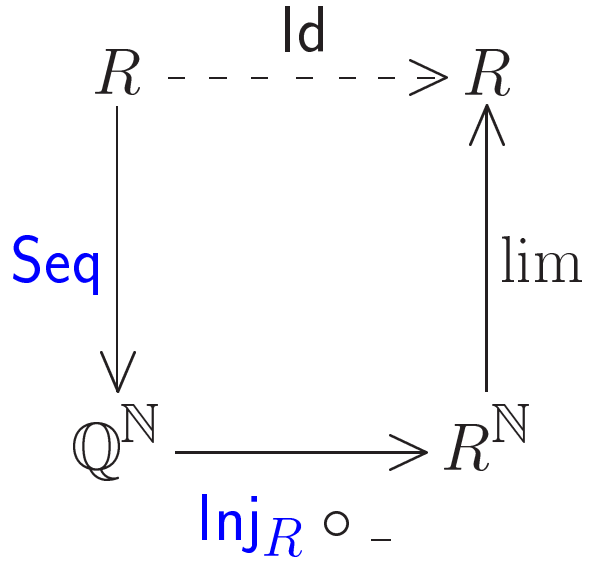
- Prove that CSeq_F is a Real Number Structure.

Categoricity of Real Number Structures:

- A morphism between R_1 and R_2 is a $\varphi : R_1 \to R_2$ that
  1. is reflects apartness: $\forall x, y : R_1.\ \varphi(x)\#\varphi(y) \to x\#y$.
  2. preserves order, addition and multiplication.
- A morphism preserves 'all' structure (Cauchy property, limits, ...)
  (Use: $\forall y \in R_2.\exists x \in R_1(\varphi(x) < y)$.)
- $R_1 \simeq R_2$ if $\varphi : R_1 \to R_2$ for some bijective morphism φ.

Categoricity of Real Number Structures ctd.:

- All Real Number Structures are isomorphic to $\mathbb{Q}^{\mathbb{N}}$:



- $\mathsf{Inj}_R : \mathbb{Q} \to R$ canonically maps rationals into R.
- $\mathsf{Seq} : R \to \mathbb{Q}^{\mathbb{N}}$: For $x : R$, define the $\mathbb{Q}$-sequences $\{q_i\}_{i=0}^{\infty}$ (increasing) and $\{r_i\}_{i=0}^{\infty}$ (decreasing), such that
  1. $\forall n{:}\mathbb{N}(r_n - q_n = (r_0 - q_0)(\frac{2}{3})^n)$
  2. $\forall n{:}\mathbb{N}(\mathsf{Inj}_R(q_n) \leq x \leq \mathsf{Inj}_R(r_n))$

Some Conclusions:

- Real mathematics, involving both a bit of algebra and a bit of analysis can be formalised completely within a theorem prover (Coq).
- Setting up a basic library and some good proof automation procedures is a substantial part of the work.

Some Conclusions:

- Real mathematics, involving both a bit of algebra and a bit of analysis can be formalised completely within a theorem prover (Coq).
- Setting up a basic library and some good proof automation procedures is a substantial part of the work.
- But it pays! See the poster by Luis Cruz-Filipe, formalizing the Fundamental Theorem of Calculus (diff $=$ int^{-1}) on top of this.

Some Conclusions:

- Real mathematics, involving both a bit of algebra and a bit of analysis can be formalised completely within a theorem prover (Coq).
- Setting up a basic library and some good proof automation procedures is a substantial part of the work.
- Computationally, the behaviour of the algorithm depends mainly on the representation of the reals.
  See the poster by Milad Niqui on efficient representations of $\mathbb{Q}$ and $\mathbb{R}$.

Some Conclusions:

- Real mathematics, involving both a bit of algebra and a bit of analysis can be formalised completely within a theorem prover (Coq).
- Setting up a basic library and some good proof automation procedures is a substantial part of the work.
- Computationally, the behaviour of the algorithm depends mainly on the representation of the reals.
- The non-determinism in the root-finding algorithm (non-continuity of the root-finding function) lies in the taking of a k-th root in $\mathbb{C}$.
- How to present a proof?? Curry Howard: proofs-as-terms Converse?? Can we present formal proofs in a readable way?

Some Conclusions:

- Real mathematics, involving both a bit of algebra and a bit of analysis can be formalised completely within a theorem prover (Coq).
- Setting up a basic library and some good proof automation procedures is a substantial part of the work.
- Computationally, the behaviour of the algorithm depends mainly on the representation of the reals.
- The non-determinism in the root-finding algorithm (non-continuity of the root-finding function) lies in the taking of a k-th root in $\mathbb{C}$.
- How to present a proof?? Curry-Howard: proofs-as-terms
  Converse: Hurry-Coward: present terms as 'readable' proofs