A New Look at the Fundamental Theorem of Algebra

Luís Cruz-Filipe

Seminário de Álgebra, Universidade de Coimbra

January 20, 2004

University of Nijmegen, The Netherlands

Centro de Lógica e Computação, Portugal

From 1.9.2004 *the University of Nijmegen will be called Radboud University of Nijmegen*

Applications

of a

Constructive

Formalization

of the FTA

Program Extraction:

- what
- why
- how

- precise notion
- why
- "natural examples"

- meaning
- why
  - theoretical interest
  - practical interest
- how...

Overview

1. Formalization of Mathematics

2. Constructive Mathematics

3. Program Extraction

4. The FTA Proof

5. Conclusions & Future Work

Formalizing Mathematics

What Faithful representation of *proofs* in a computer

Why High guarantee of correctness

Presentation and exchange

Applications

Constructive Mathematics

Intuitionistic logic (Brouwer): do not accept $A \vee \neg A$

$\rightsquigarrow$ no clear interpretation of this axiom

$\rightsquigarrow$ realizability: proof of $\forall x.\exists y.P(x,y)$ defines a (computable) function

Natural examples:

- Curry–Howard isomorphism
- internal logic of a topos

Program Extraction

Idea: make the implicit algorithm in a proof of $\forall x.\exists y.P(x, y)$ explicit

$\rightsquigarrow$ distinction between the actual *algorithm* and its *properties*

$\rightsquigarrow$ proofs may influence results of computations (e.g. $\frac{1}{x}$)

Useful when correctness is more dear than efficiency

The Fundamental Theorem of Algebra

Theorem. Let f be a non-constant polynomial with complex coefficients. Then f has a root, i.e., there exists a complex number z such that $f(z) = 0$.

Proof. [H. Kneser, 1940] Let f be a polynomial over $\mathbb{C}$. Then $|f(z)| \longrightarrow \infty$ as $|z| \longrightarrow \infty$, therefore $|f|$ has a minimum at $z_0 \in \mathbb{C}$.

Take $g(z) = f(z - z_0) = \sum_{i=0}^{n} a_n z^n$ and suppose $g(0) \neq 0$. Take the least $k > 0$ s.t. $a_k \neq 0$; then $g(z) = a_0 + a_k z^k + O\left(z^{k+1}\right)$.

Taking ε small enough, at $z' = \varepsilon \sqrt[k]{-\frac{a_0}{a_k}}$ the term in $O\left(z^{k+1}\right)$ will be negligible, and $|g(z')| \approx |a_0|(1 - \varepsilon^k) < |g(0)|$. Contradiction.

The Fundamental Theorem of Algebra (cont.)

$\rightsquigarrow$ not a constructive proof: we prove $\neg\neg\exists z. f(z) = 0$, which is weaker than the intended $\exists z. f(z) = 0$.

$\rightsquigarrow$ however, given z such that $|f(z)| > 0$ the proof contains a construction of z' with $|f(z)| > |f(z')|$

$\rightsquigarrow$ might this be used to define a Cauchy sequence converging to a root of f?

Problem: conflicting demands on ε

The FTA for Monic Polynomials

Three problems:

1. equality not decidable;
2. choosing k with $\sqrt[k]{\frac{|b_0|}{|b_k|}}$ minimal not possible;
3. taking k_j with $\left|b_{k_j}\right| r_j^{k_j}$ maximal not possible.

The FTA for Monic Polynomials (cont.)

To solve (1): restate the result as

"if $|f(z_i)| < c$ then $|f(z_{i+1})| < qc$"

with q as above.

Now we can decide whether $|f(z_i)| < qc$ or $|f(z_i)| > 0$, and the proof can proceed as before.

The FTA for Monic Polynomials (cont.)

To solve (2): take a minimum "up to ε"; that is, simultaneously define r_0 and k_0 such that, given $\varepsilon > 0$,

$$\begin{aligned} a_{k_0} r_0^{k_0} &= a_0 + \varepsilon \\ a_i r_0^i - \varepsilon &< a_{k_0} r_0^{k_0} \end{aligned}$$

(Start with $k_0 = n$, $r_0 = \sqrt[n]{a_0 - \varepsilon}$.
For each i down to 1:
 – if $a_i r_0^i < a_0$ do nothing;
 – if $a_i r_0^i > a_0 - \varepsilon$, redefine $k_0 = i$ and $r_0 = \sqrt[i]{(a_0 - \varepsilon)/a_i}$.
When i reaches 0, k_0 and r_0 will satisfy the above conditions.)

The Fundamental Theorem of Algebra: General Case

Idea: given $f(z) = \sum_{i=0}^{n} a_n z^n$, apply the previous result to f/a_n.

Problem: even if f is non-constant there is no guarantee that $a_n \neq 0$.

$\rightsquigarrow$ different approach, proof by induction on n.

The Implicit Algorithm Made Explicit

$\rightsquigarrow$ M. Kneser's original proof corresponds to the Newton–Raphson algorithm to find a root of the polynomial

$\rightsquigarrow$ the version presented (and formalized) is slightly less efficient, because k_i's start at 0 instead of -1

$\rightsquigarrow$ currently, basic arithmetic too slow; computation of square roots in $\mathbb{R}$ takes too long

Conclusions & Future Work

- Formalizing mathematics is useful
- “Forgetting” the principle of the excluded middle not too dramatic
- Program extraction may one day be “right” way to program