doi:10.20396/revpibic262018600

XXVI Congresso de Iniciação Científica Unicamp
17 a 19 de outubro Campinas | Brasil

Formalization of Mathematics Through Proof Assistants: A study on the state of the art

Henrique Y. R. Inonhe*, Walter Alexandre Carnielli

Abstract
The formalization of mathematics in practice relies heavily on proof assistants and automatic theorem provers, therefore we studied what are the state of the art proof assitants and their limitations to understand what are the main challenges in making formalized mathematics common practice among mathematicians. We found out that curretly the two major dificulties in formalizing mathematics with proof assistants are due to steep learning curves in how to use these tools and due to a wide gap between the notation employed in these proof assistants and the currently used mathematical notation. We also developed a C++ library to develop proof assistants with great notational flexibility.

Key words:
Proof assistants, formalization of mathematics, philosophy of formal sciences.

Introduction

The formalization of mathematics consists in representing mathematical statements and proofs in a formal axiomatic system in such a way that the correctness of these proofs can be verified mechanically, that is, the verification process is algorithmic and can be done by a computer, without resorting to creativity or intuition. However, the formalization process is cumbersome and cannot be carried out in practice “by hand”, therefore we must resort to specialized computer tools such as proof assistants to do the job.

There are three major benefits in this formalization process:
1. To make mathematics more precise and trustworthy by delegating proof verification to computers instead of human referees, while also making the verification process much more efficient and less time consuming.
2. The development of a (possibly unified) database of mathematical proofs and theorems altoghether with tools to navigate through these results.
3. An enhanced understading of mathematical proofs, given that in a formal proof there can be no inference jumps nor unstated assumptions.

Results and Discussion

Initially three very mature and widespread proof assistants were chosen to be studied, namely Mizar, Isabelle and Coq.
Both Isabelle and Coq use an imperative proof style in which the user inputs inference rules and tactics (usually goal oriented in a backwards reasoning fashion).
Mizar on the other hand uses a declarative proof style which is more natural and is closer to the current mathematical practice.
Isabelle also has a declarative proof mode called Isar, which resembles Mizar proof style.
Isabelle and Coq both support user defined tactics and have very powerful proof automation systems, which is a double edged knife, in the sense that while making proofs easier to do, automation also makes them less descriptive and harder to understand.
Isabelle has a very interesting annotation system that makes it possible to reintroduce operators in a mixfix notation.

Isabelle was chosen to be studied more in depth and to better understand the way it works I formalized some Propositional Logic and elementary set theoretical theorems in it.

```
theorem
  assumes "prime (p::nat)"
  shows "sqrt (real p) ∈ \<rat>"
proof
  from `prime p` have p: "1 < p" by (simp add: prime_nat_def)
  assume "sqrt (real p) ∈ \<rat>"
  then obtain m n :: nat where
      n: "n ≠ 0" and sqrt_rat: "¦sqrt (real p)¦ = real m / real n"
    and gcd: "gcd m n = 1" by (rule Rats_abs_nat_div_natE)
  from n and sqrt_rat have "real m = ¦sqrt (real p)¦ * real n" by simp
  then have "real (m²) = (sqrt (real p))² * real (n²)"
    by (auto simp add: power2_eq_square)
  also have "(sqrt (real p))² = real p" by simp
  also have "… * real (n²) = real (p * n²)" by simp
  finally have eq: "m² = p * n²" ..
  then have "p dvd m²" ..
  with `prime p` pos2 have dvd_m: "p dvd m" by (rule prime_dvd_power_nat)
  then obtain k where "m = p * k" ..
  with eq have "p * n² = p² * k²" by (auto simp add: power2_eq_square mult_ac)
  with p have "n² = p * k²" by (simp add: power2_eq_square)
  then have "p dvd n²" ..
  with `prime p` pos2 have "p dvd n" by (rule prime_dvd_power_nat)
  with dvd_m have "p dvd gcd m n" by (rule gcd_greatest_nat)
  with gcd have "p dvd 1" by simp
  then have "p ≤ 1" by (simp add: dvd_imp_le)
  with p show False by simp
qed
```

Figure 1. An Isar proof of the irrationality of the square root of 2 in Isabelle.

Trying to address the need of improvement in notation in formalized mathematics as well as making it closer to mathematical notation found in textbooks I developed an open source C++ library for developing proof assistants, that I hope will help people who are interested in the subject develop their own proof assistants for any preferred logic and/or deductive system.

Conclusions

Currently, the two major challenges for a widespread formalized mathematics are making proof assistants easier to use (specially for mathematicians) and improve the presentation and readability of proofs making them closer to hand-written proofs.

Acknowledgement

I would like to thank my advisor Walter Carnielli, CLE (Centro de Lógica, Epistemologia e História da Ciência) and CNPq.