Mathematics with Coq for first-year undergraduate students

Pierre Rousselin*

Université Paris 13, Villetaneuse, France
rousselin@math.univ-paris13.fr

Abstract

We have experienced teaching mathematics with Coq to mathematics and computer science students during the first semester of their first university year for two years. This talk is about the content of this hands-on course, its reception by the students and some perspectives and questions for a broader (in scale and content) course.

1 Introduction

During fall 2021 at Université Paris 13, we have created with Marie Kerjean and Micaela Mayero a new course named "Introduction to Formal Proofs" for first year double-major computer science and mathematics students. Its main goals are:

- to challenge the students with a new subject involving both mathematics and computer science;
- to teach rigour in mathematics and computer science;
- to make them very active in their research for a proof.

The course is mandatory for these double-major students. It uses the Coq proof assistant and is completely hands-on, with no dedicated lecture hours and has involved each year two groups of roughly 25 students each during 18 hours divided as 6 practical sessions. In addition to these sessions, students had to complete two homework assignments and take a $1h30$ exam.

The Coq proof assistant was chosen because the author (who did not have any experience with proof assistants) was helped by Micaela Mayero and Marie Kerjean who used it in their research. The students used CoqIDE as text editor and, in addition to our local (Debian packaged) installation, easily installed Coq and CoqIDE on their own computers using links to the Coq Platform. The content of the course, as Coq source files, is available on the author's web page[1], but it is in French, and requires more polishing.

2 Summary of the course

The course is based on Coq's standard library and has the following syllabus:

1. first steps in constructive propositional logic
2. Peano's natural numbers and induction
3. predicate calculus, existential quantifier
4. Real numbers and analysis:
   (a) $\mathbb{R}$ as a field; proving equalities
   (b) $\mathbb{R}$ as an ordered field, proving inequalities
   (c) absolute value and usual distance on $\mathbb{R}$

*with the participation of Marie Kerjean and Micaela Mayero (LIPN, Université Paris 13)
[1] https://www.math.univ-paris13.fr/~rousselin/ipf.html

(d) convergence of sequences of real numbers

The first two parts have been well received by the students, although the constructive approach is different from what they learn (at the same time) in a classical "pen and paper" mathematics course. Learning logic *via* introduction and elimination rules (although this was not formally stated) is certainly closer to a regular mathematical activity than truth tables. Intuitionist logic is pedagogically sound in that it often forces the students to reason in a very directed way; for instance to prove an implication, we *have* to assume (`intros`) that its antecedent holds. The visual and playful aspect of Coq certainly played a part here. Of course, Coq shines when it comes to induction, giving the students a firm and clear setting to write their proofs.

The next topic (predicate calculus) is usually difficult for students. The existential quantifier is, in some sense, new to them. Of course, in high school, there are existential statements (such as "there exists a unique differentiable function from $\mathbb{R}$ to $\mathbb{R}$ which is equal to its derivative and has value 1 at 0"), but they are either not proved or proved using unclear proof schemes. In this part, as in many others, Coq poses no further difficulty but acts as a magnifier of the students lack of understanding. Again, restricting oneself to intuitionist logic, at least in the beginning, gives a very clear way of using and proving existential propositions.

The last part is based on (the "classical" part of) the `Reals` sublibrary of Coq's standard library. It was created more than twenty years ago by Micaela Mayero ([2]). While it certainly needs some polishing, it has strong arguments for such a course: it is simple (there are not many visible abstractions which could obfuscate the basic lemmas); it is standard, hence comes with any installation of Coq; it was originally based on (more or less) standard axioms, and remains organized with 16 lemmas as primitives, which corresponds to a standard analysis course at this level.

The aim of the last part is to go from algebra to real analysis. The students are encouraged to build everything from the primitive lemmas. Powerful tactics like `lra` and `ring` are introduced during the last session, in conjunction with a more "forward" style, when students have to prove, for instance, that the sum of two convergent sequences is convergent. To be honest, only the best students have been able to finish the part about numerical sequences, but most of them have covered an impressively large amount of content in such a small time.

3 Conclusion and further work

We have taught first-year Computer Science and Mathematics students how to write proofs of mathematical statements in Coq. Our course is certainly not perfect but we believe that it helped them grasp some very important ideas and proof schemes. At this level, most of the students' difficulties originate from mathematics, not from Coq which, of course, has not the magical power to make mathematics easy. We cannot, at this stage, precisely quantify how our course helped the students in their other courses, but their further results went in the right direction (their final average grade for this year was around 5 points out of 20 above the other students but there is a strong selection bias).

While our course does not at the moment use more recent additions to Coq, it would probably not have been possible without the huge pedagogic work done for the electronic book [3]. We mostly followed its method (everything is written in Coq source files, we always introduce new tactics with examples and follow these with easy exercises) and adopted its reduced set of tactics.

While we did not in such a small course worked directly on how to translate formal proofs into pen-and-paper proofs, we believe that using a proof assistant to work on reasoning and proofs has many benefits: it prevents the students to write nonsense and helps them with a clear view of the proof state. Also, at least in the beginning, working with intuitionistic logic gives a natural direction to proofs. However, we did not use that much the computation power of Coq and this course could probably be translated into Lean or Isabelle/HOL without much effort.

At the same time we are aware that we chose a particularly convenient mathematical subject. At this basic level, numerical sequences do not need a lot of abstraction. Even the study of numerical functions is made a lot more complicated by questions like partiality or the pervasiveness of intervals. In the future, we

would like to extend our material with more content from classical mathematics courses in first year (even though this is outside the scope of this small course). This includes, in particular:

- classical theorems in analysis of continuous numerical functions : intermediate value theorem, ...
- differentiation of numerical functions, asymptotic analysis
- finite dimensional vector spaces

The question of the underlying logic will also need to be settled: at this moment, the logic of "classical" `Reals` is stronger than intuitionist logic but weaker than classical logic (it assumes the weak excluded middle and the limited principle of omniscience, see [4] or the source file https://github.com/coq/coq/blob/master/theories/Reals/ClassicalDedekindReals.v). Do we decide for our courses to move completely to classical logic? If so, should we add new tactics? (For instance, Lean's mathlib has `push_neg`). Do we further assume functional extensionality (already used for Dedekind cuts in the construction of the "classical" real numbers, again see [4])? If so, do we use tools to rewrite under binders (as is common in mathematics)?

Finally, this effort is part of a general movement: the usage of proof assistants in first year of university is gaining momentum, see [1].

References

[1] M. Kerjean, F. Le Roux, P. Massot, M. Mayero, Z. Mesnil, S. Modeste, J. Narboux, and P. Rousselin. Utilisation des assistants de preuves pour l'enseignement en L1. *Gazette de la Société Mathématique de France*, octobre, 2022.

[2] Micaela Mayero. *Formalisation et automatisation de preuves en analyses réelle et numérique*. PhD thesis, Université Paris VI, décembre 2001.

[3] Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey. *Logical Foundations*, volume 1 of *Software Foundations*. Electronic textbook, 2023. Version 6.3, http://softwarefoundations.cis.upenn.edu.

[4] Vincent Séméria. Nombres réels dans coq. *Actes des 31es Journées Francophones des Langages Applicatifs (JFLA)*, pages 104–111, 2020.