Toward the use of a proof assistant to teach mathematics

Julien Narboux

TOWARD THE USE OF A PROOF ASSISTANT TO TEACH MATHEMATICS.

Julien Narboux
Ecole Polytechnique, France
Julien.Narboux@inria.fr

Proof is a crucial aspect of mathematics and must have a prominent role in the education. Dynamic Geometry Software (D.G.S.) and Computer Algebra Software (C.A.S) are widely used in a pedagogical context. These tools are used to explore, visualize, calculate, find counter examples, conjectures, or check facts, but most of them can not be used to build a proof in itself. But there are software whose sole purpose is to help the user produce proofs : the ***proof assistants****. We believe that proof assistants are now mature enough to be adapted to the education. After giving a quick overview of what a proof assistant is, we will discuss the possible advantages of using it in the education. Finally we report on the ongoing work to ease the use of a proof assistant in the classroom.*

Introduction

Technological tools are widely used to teach mathematics in schools. Dynamic Geometry Software (D.G.S.) and Computer Algebra Software (C.A.S) are the two families of software that are well represented. These tools are widely used to explore, experiment, visualize, calculate, measure, find counter examples, conjecture... but most of them can not be used directly to build a proof. It is not very difficult to find a D.G.S, after a few minutes search on the Internet one can build this list of systems (not meant to be exhaustive): Cabri Géométre, CaR, Cinderella, Déclic, Dr. Geo, Euclid, Euklid, DynaGeo, Eukleides, Gava, GeoExp, GeoFlash, GeoLabo, Geometria, Geometrix, Geometry Explorer, GeoPlanW, GeoSpaceW, GEUP, GeoView, GEX, GRACE, KGeo, KIG, Non-Euclid, Sketchpad, Xcas...
Among these software some can in some sense deal with proofs: for instance GEX

can produce proofs using automated deduction methods, Xcas allows the user to build an analytic proof using its C.A.S capabilities and Cinderella can act as an oracle checking a fact using a probabilistic method.
The only D.G.S. we found which can be used to build proofs interactively is Geometrix. It allows the student to prove assertions using Ia base of known lemmas.
This situation is quite surprising because proof is a crucial aspect of mathematics as shown for example by the recent changes in the French mathematics curriculum which emphasize the prominent role of proof. Moreover proof is by definition (or at least in its formal definitions) checkable mechanically.
However the status of proof in the education and the impact of dynamic geometry software on the teaching of geometry and more precisely on the proving activity is a well addressed issue in the literature (See for example Furinghetti, F. & Paoloa Domingo (2003), de Villiers (1990)).
In Furinghetti, F. & Paoloa Domingo (2003), the authors “confirm the interlacement of exploration and proof”. We think that these two activities could better interlaced if they were both conducted using the computer. But because the proving activity is a creative activity, we believe that an ad-hoc geometry prover as Geometrix is not sufficient to give to the student enough freedom to prove its statements the way he wants to (the student may want to use an analytic method, or define a new lemma for instance).
Systems have been designed in the sole purpose of proving theorems. These software called proof assistants have been designed by people working in the proof theory and logic communities. I believe that proof assistants are now mature enough to be adapted to be useful in the education.

What is a proof assistant ?

A proof assistant is a software which can be used to interactively build a formal proof. This object is then mechanically checked by the machine. There are many proof assistants, we can cite for instance Coq, PVS, Isabelle, Mizar, HOL,... In each of these systems the user has to first define what he wants to prove and then guide the software until it accepts the proof as a valid proof. Proof assistants are different from theorem provers. Proof assistants are designed for an interactive use whereas theorem provers a build to find proofs automatically. However proof assistants often provides ways to automate the production of some proofs. They implement decision procedures for some languages (for example propositional calculus), but their main concern is not automation in itself. Automation is just a way to ease the proof process. Proof assistants such as Coq have been used to formally prove theorems and programs. We can cite among the best achievements the formalization of the proof of the four colours theorem by Georges Gonthier and Benjamin Werner (see Gonthier (2004)) and the proof of the fundamental theorem of algebra by the group of Henk Barendregt.

Can we trust a proof checked by the Coq Proof assistant ?

If a proof created by a student is accepted by the Coq proof assistant the teacher can have a very high level of confidence in this proof. In fact, to trust a proof check by the Coq proof assistant you only need to trust :

- **The theory behind Coq**

  The theory of Coq version 8.0 is generally admitted to be consistent with regard to Zermelo-Fraenkel set theory + inaccessible cardinals. Proofs of consistency of subsystems of the theory of Coq can be found in the literature.
- **The Coq kernel implementation**

  You have to trust that the implementation of the Coq kernel mirrors the theory behind Coq. The kernel is intentionally small to limit the risk of conceptual or accidental implementation bugs.
- **Your hardware, operating system and Objective Caml Compiler**

  In theory, if your hardware or operating system does not work properly, it can accidentally be the case that False becomes provable. But it is more likely the case that the whole Coq system will be unusable. You can check your proof using different computers if you feel the need to.
- **Your axioms**

  Your axioms must be consistent with the theory behind Coq.

Why should we use a proof assistant in the classroom ?

Flexibility: Compared to an ad-hoc geometry prover a proof assistant provides a way to give to the students as much freedom as he would have using a hand written proof while still maintaining correctness. The students should have both the ability to make arbitrarily complex proofs, define new notions or lemmas and use a base of known lemmas prepared by the teacher.

Rigour: It is often not very clear for a student to know when a proof is correct and what is the level of details needed for a proof to be acceptable. In fact, it often depends on the teacher and mathematical subject. For example the proofs of geometry theorems produced by student in high-school are usually much closer to a formal proof than proofs concerning properties of functions or probabilities.
The use of a proof assistant gives a clear and fair notion of what a correct proof is: A proof is correct if it is accepted by the computer.

Clarity: Experience shows that many students do not have a very clear view of what is a proof. We believe that the experience of doing a formal proof would help the student understand what is a proof as the use of a proof assistant clarifies these questions:

- What is a definition, a theorem, an axiom ?

  Each statement of the formal development is labelled by its status, it is always clear what are the definitions, lemmas, axioms and admitted theorems.
- What are the logical rules involved ?

  Each step of the formal proof is basically an application of a logical rule.
  The user can for example apply a theorem, assert a new fact, decompose the conjunction of two propositions into two separated propositions, start a proof by induction, start a new lemma, ...
- What are the hypotheses and the goal ?

  The proof assistant provides the list of the hypotheses and the list of the statement which remain to be proved.

Interactivity: At each step of the demonstration the proof assistant check if the proof is still correct, for example if the student wants to apply the Pythagoras theorem, the proof assistant checks that the required hypotheses has been shown.

What should we do to ease the use of a proof assistant in the classroom ?

Automation: Formalizing mathematics using a proof assistant is a tedious work. The user has to give to the proof assistant all the details of the proof. A formal proof is usually many times longer than an informal one.

This issue is well known in the community and proof assistants have now features to ease the automation. Automation means that some steps in the proof can be proved automatically. These features should be adapted to the pedagogical context. Indeed when the software is used by mathematicians the goal is to automate as much as possible but if we want to use proof assistants in the classroom the steps which are automated should depend on the context. For example, for advanced student the teacher may want to hide steps which consist only in simplifications of arithmetic expressions but for younger students he may want to disable this automation.

Another example is the treatment of degenerated cases while formalizing geometry.

Informal proofs often forget degenerated cases (particular cases, when two points are equals for instance), but theses cases can not be forgotten in the mechanically checked proof. These cases could be dealt with by automatic deduction in geometry methods.

Error messages: The error messages which are displayed have not been designed to be understood by a student. The error messages could be improved, when for example a theorem can not applied, the software could explain why (what is the missing hypotheses for example).

Interface: The syntax and interface for proof assistants is not as nice as the one used by C.A.S.. Usually there is no “pretty printing” is provided. This question should be addressed by the formal proofs community if we want proof assistants to be used in the classroom.

Ongoing work in the community:

Some people in the formal proofs community are working on the development of tools to ease the use in the classroom. Frédérique Guilhot has created a very large Coq development containing the main concepts and theorems used in French high school geometry (see Frédérique Guilhot (2004)). In cooperation with Yves Bertot and Loïc Pottier (2003), they have designed an interface between the Coq proof assistant and the GeoPlan dynamic geometry software (GeoView) which can display a Coq statement representing a geometry theorem.
Another ongoing work is by the author of this article, the project is to create a D.G.S. called DrGeoCaml which can not only display a statement but also be used to state a theorem and build a proof of the statement interactively.

Conclusion

We argue in this paper for the use of proof assistants in the teaching of mathematics and discuss the current issues. The main goal of this paper is to initiate the collaboration between the formal proofs community and the technology in mathematics teaching community.

References :

- Y. Bertot, F. Guilhot, L. Pottier. (2003). Visualizing Geometrical Statements with GeoView, *in the proceedings of UITP*, Rome, Italy.
- The Coq development team. (2004), The Coq proof assistant reference manual, *LogiCal Project*, Version 8.0.
- Furinghetti, F. & Paoloa Domingo. (2003). To produce conjectures and to prove them within a dynamic geometry environment: a case study. *Proceedings of PME 27: Psychology of Mathematics Education 27th International Conference*, 2 (pp.397-404). Honolulu, U.S.A.
- Frédérique Guilhot. (2004). Formalization of high school geometry in Coq.
- Julien Narboux. (2005). When a GUI for dynamic geometry becomes an interactive proof language, *High Level Languages for Proofs Workshop*, Chambery.
- Georges Gonthier. (2005). A computer checked proof of the four colour theorem.
- de Villiers, M. (1990). The role and function of proof in mathematics. *Pythagoras*, n. 24, pp. 17-24.
- Oleksiy Yevdokimov. About a constructivist approach for stimulating students' thinking to produce conjectures and their proving in acive learning of geoemetry.