OPENING A DISCUSSION ON TEACHING PROOF WITH AUTOMATED THEOREM PROVERS

GILA HANNA, XIAOHENG (KITTY) YAN

The article argues that there is a need for new approaches to teaching proof, ones that capitalize not only on newly available technology but also on modern theories of teaching and learning. It contributes to filling this need by opening a discussion on digital proof assistants, programs that allow one to do mathematics with the aid of a computer, and more specifically to construct proofs and check their correctness. The article starts by introducing and exploring such proof assistants, then goes on to argue that they could play an important role in fostering students' appreciation and understanding of proof and of mathematics as a whole. Finally, the article points out the need to develop explicit pedagogic strategies tailored to assist teachers in deploying such computer-based tools.

During the past three decades, the practice of mathematics has adopted new and interesting types of proof and argumentation that challenge previous norms. This development was triggered primarily by the increasing use of computers in mathematical research, both as heuristic devices and as tools of verification. One major outcome has been the creation and acceptance among practicing mathematicians of what might be referred to broadly as computer-assisted theorem development, in which mathematicians make use of both interactive proof assistants and fully automatic proving software (see Hanna, Reid & de Villiers, 2019).

This change in mathematical practice, however, has not been reflected, by and large, in mathematics teaching. A few universities have offered courses on the use of digital proof assistants, but undergraduate mathematics curricula have neglected them. As a result, there is no solid evidence for the degree to which proof assistants in the undergraduate classroom might help students construct and understand valid proofs. In fact, there does not appear to be any published systematic research that has explored their potential in any educational context.

Theorem provers in mathematical practice

Among the first uses of proof assistants was the machine-assisted translation of informal proofs into formal ones, which opened the door to machine-assisted verification. Today the correctness of a proof, once formalized, can be verified to a level of certainty that mathematicians cannot achieve when working with the informal proofs that have long been their stock in trade (Avigad & Harrison, 2014; Voevodsky, 2014).

While in their early stages these programs could assist only with verifying the correctness of existing proofs or formulating and assessing conjectures, their greatly expanded capabilities now allow mathematicians to take the next step and employ them in actually creating a new proof. These new capabilities were once difficult to exploit because they required specialized computer skills, but they have now become increasingly accessible to experts and novices alike.

Automated proof assistants can be classified into two main types based on their capabilities and accordingly on how they are used in creating proofs. Interactive theorem provers (ITPs) facilitate the construction of proofs through close human-machine collaboration, while automated theorem provers (ATPs) [1] are designed to prove mathematical theorems with a very minimum of human intervention.

The main function of an ATP is to construct a formal proof that is valid within a given logical system. In other words, an ATP is designed to accept a set of axioms and rules of inference and then to construct a chain of inferences that shows how a given conjecture or other statement provided to it (and itself well formed in the system) is a logical consequence of the axioms. In pursuing this goal an ATP is also capable of deciding that it cannot prove the given conjecture or statement because it is not compatible with the axioms, and may also identify a counterexample [2].

Formalizing proofs with an automated theorem prover

Informal proof—sometimes incorporating formal passages—has long been the standard in mathematical practice, though widely recognized as open to error. As Ganesalingam and Gowers (2017) have written, "there is plenty of evidence that incorrectness pervades the published literature" (p. 254). But most mathematicians were prepared to accept the possibility of error, because they simply did not want to make the formidable extra effort required to formalize their proofs. They found it more reasonable to settle for informal proofs, which also offered the advantage of being understandable and thoughtfully checkable.

A formal proof demands a detailed derivation, in which each statement in the chain, other than the axioms, follows from preceding statements by explicit rules of inference. These rules must be made explicit at the outset, and must be cited whenever used at any step of the derivation. And, of course, all statements must be well formed, conforming to explicit rules. Such a derivation may have the great advantage of assuring the validity of an inference, but it is cumbersome to create, tiresome for others to follow, and, perhaps most importantly, often makes it much more difficult to grasp the mathematical ideas at play.

However, a formal proof does have another advantage, in that it allows mechanical checking of its correctness. For this reason, too, the advent of automated theorem provers has been of great interest to the many mathematicians who saw the prospect of more thorough verification of their own research results. Mathematicians are often not confident their results are correct, but have avoided formalization because it is too difficult and time-consuming to do by hand. Automated theorem provers have come to their rescue by offering the prospect of automated formalization.

Automated formalization, unlike the rare formalization by hand, also offers a manageable way to address the widespread perception that the conventional refereeing process—based on informal proofs—is far from adequate. In the view of Hales (2008), "When the part of refereeing a mathematical article that consists of checking its correctness takes more time than formalizing the contents of the paper would take, referees will insist on getting a formalized version before they want to look at a paper" (p. 1414). Harrison (2008) too embraces the formalization of mathematical arguments, saying that, "the traditional social process is an anachronism" (p. 1400). He adds that once mathematicians are assured of the correctness of their formally verified results, they can then present them to others "in a high-level conceptual way […] in principle, a computer program can offer views of the same proof at different levels of detail to suit the differing needs of readers" (p. 1400).

Expanded capabilities

Computer scientists have expanded the capabilities of proof assistants to the point that mathematicians can use them to construct an entire proof for a given conjecture, with varying degrees of human intervention. Both ATPs and ITPs can now be used in exploratory mode, letting mathematicians more easily develop new conjectures. Thus mathematicians can now work closely with an automated proof assistant to explore avenues of reasoning as well as different and perhaps unexpected steps to a satisfactory proof (Bundy, 2011) [3].

Significantly, computer scientists have also worked to improve the readability of ATP output and the options available: "This points to an important new direction for automated reasoning: multi-level proof presentation in which the user can choose the level of granularity of the proof and which highlights the key ideas and the hard parts of the proof" (Bundy, 2011, p. 9). In the same vein, there have also been attempts to design ATP output to be closer to that of a human being. Ganesalingam and Gowers (2017) describe a type of ATP that provides help to mathematicians in a particularly friendly manner. Although they did not expect its output to be identical to that of a human, they report that: "despite this, the program did reasonably well at fooling people that it was human" (p. 287).

Acceptance

There is now no shortage of automated proof assistants, addressing the many diverse aspects of proving. The mathematician Emily Riehl finds them very useful, saying, "It's not necessarily something you have to use all the time […] but using a proof assistant has changed the way I think about writing proofs" [4]. Their reception, however, has been decidedly mixed. Ganesalingam and Gowers (2017) point out that, "these days there is a thriving subcommunity of mathematicians who use interactive theorem provers such as Coq, HOL, Isabelle and Mizar" (p. 254), but they concede that, "it is also noticeable that the great majority of mathematicians do not use these systems and seem unconcerned about the possibility that their arguments are incorrect" (p. 254).

The mathematician Michael Harris is among those who do not use interactive theorem provers and maintain they are not necessary. He estimates that it is not worth his while to invest time in learning the instructions needed to communicate with the automated prover, believing that "by the time I've reframed my question into a form that could fit into this technology, I would have solved the problem myself" [4].

There are no published statistics on what percentage of practising mathematicians make use of automated proof assistants and to what extent. Nevertheless, the existence of many international conferences, workshops, and scholarly journals on automated theorem proving indicate that these systems are widely used and are attracting attention.

Theorem provers in mathematics education: opportunities

As discussed, new technology is changing the way mathematicians work, but its reflection in mathematics education has been limited to specific subjects. Mathematics educators have already garnered a great deal of experience using digital tools to teach geometry, algebra, and statistics, and have found them very valuable. But this success has not carried over to the subject of proof. The time has come to introduce digital proof assistants to the mathematics classroom and to investigate the opportunities these state-of-the-art tools might provide to foster enhanced understanding of proof in particular and of mathematics as a whole.

There is also a broader context. Because digital technologies are critical for success in our knowledge-based society, both schools and universities are under increasing pressure to produce graduates who understand technology and are able to integrate it into their work. To help schools and universities use technology to better understand content, the recent paper 'Transforming the mathematical practices of learners and teachers through digital technology' (Hoyles, 2018) advocates a long-term program of research and development in mathematics education, and in particular the creation of a solid base of knowledge on effective instructional practices.

First steps

Some twenty years ago, when proof technology was not as advanced and as friendly as it is today, Melis and Leron (1999) were among the first to advocate its use in teaching proof. They argued that digital technologies would encourage students to engage in proof planning, and that this in turn would support the active learning of theorem proving. This is because, in their judgment, the technology would help students explore a problem interactively and thus single out the particular tasks and skills required for the proof. In the authors' estimation, the technology could also help learners of proof by providing an intelligently designed pre-selection and ordering of methods, as well as by keeping a record of their proof attempts. As far as we know, this early proof technology teaching advice has not been implemented in mathematics education to any extent.

In the last five years or so, readily available proof assistants have become used more broadly in teaching mathematics and logic at the undergraduate level. Some recently published papers have implied that an automated proof assistant can help students learn to prove. The theorem prover Lean, for example, has been used successfully in undergraduate classes.

The Lean theorem prover

Only a few universities have introduced automated proving at the undergraduate level. At Carnegie Melon University, for example, Jeremy Avigad teaches logic and proof with the help of the interactive theorem prover Lean [5]. At Imperial College Kevin Buzzard is working with a team of first-year undergraduates, teaching them to use it to prove theorems. The last edition of the Lean textbook [6], states that one of the goals of the Lean theorem prover is to support not only proving but mathematical reasoning in general. The authors also say that Lean "aims to bridge the gap between interactive and automated theorem proving, by situating automated tools and methods in a framework that supports user interaction and the construction of fully specified axiomatic proofs" (p. 1).

Avigad (2019) recently described a 14-week (semester) course called 'Learning logic and proof with an interactive theorem prover' that he has been teaching to undergraduate classes. It makes use of the Lean theorem prover to provide an introduction to mathematical proof, symbolic logic, and interactive theorem proving. This course is considered appropriate for first- or second-year undergraduate students and has been taught for the last three years. Its four goals are:

> to teach [students] to write clear, literate, mathematical proofs; to introduce them to symbolic logic and the formal modeling of deductive proof; to introduce them to interactive theorem proving; to teach them to understand how to use logic as a precise language for making claims about systems of objects and the relationships between them, and specifying certain states of affairs. (p. 279)

Part of the instruction in this course is focussed on showing students "how to write formal expressions and proofs that can be checked automatically" (p. 279). One third of the weekly homework exercises required the use of Lean, while the other two thirds relied on conventional methods. The instructor reported noticeable increases in student motivation and achievement, and students gave the course high evaluation scores.

In teaching this course, Avigad found that using both formal and informal language led students to reflect on the power of symbolic language to deliver more precise information and to shed light on ambiguities present in natural language. He also noted "anecdotal evidence to support the claim that interactive theorem proving software helps teach students mathematics" (p. 290). It had not been his intention to undertake the systematic evaluation that would be needed to lend more weight to this assertion.

In his 2019 article in *Motherboard Magazine* [7], Rorvig reports that Kevin Buzzard now uses automated theorem provers enthusiastically in teaching undergraduates. Rorvig adds that "When Buzzard started using the proof verification software called Lean, he became hooked. Not only did the software allow him to verify proofs beyond any doubt, it also promoted thinking about math in a clear and unmistakable way." Buzzard says he believes computers can help mathematicians and students construct proofs, and furthermore that he can help make this happen sooner "by (a) helping to build a database of modern mathematical theorems and definitions and (b) trying to teach mathematics undergraduates how to use the software" [8].

It must be said, however, that the observation that the software "also promoted thinking about math in a clear and unmistakable way" was not backed at the time by structured empirical data. In fact, there do not appear to be any research papers reporting on the extent to which Lean supports students in thinking about proof, making choices, defining their goals, making design decisions, and evaluating their progress. Nor are there research papers assessing the role of the teacher as the provider of support for the efficient use of Lean, or recommending effective instructional approaches.

Lean now has an online forum, Zulip, and a large community of enthusiastic users [9]. It is actually under continuous development, as users fine-tune it and add new features.

GeoGebra's automated proving tools

GeoGebra is an open-source dynamic mathematics program designed for the teaching and learning of mathematics from middle school through the undergraduate level. It has gained in popularity over the last twenty years and is now widely used. GeoGebra has recently added an Automated Reasoning Tool (ART) to help students conjecture that a given property holds for a specific geometric object and then to find a proof that their conjecture is true. If that is not the case and the property does not hold, ART can also help students make the necessary changes to the original conjecture (Hohenwarter, Kovács & Recio, 2019, p. 216).

Since the developers of GeoGebra added reasoning tools to their software, they have published a large number of papers in scholarly journals describing the potential of those tools for secondary-school learning (see, *e.g.,* Botana, Hohenwarter, Janičić, Kovács, Petrović, Recio & Weitzhofer, 2015). These additions appear to benefit students at both the undergraduate and the secondary level. Botana *et al.* (2015) say that the enhancements to GeoGebra were made with five goals in mind:

1. To provide an intuitive interface
2. To allow for simplified output
3. To increase program execution speed
4. To make GeoGebra's subsystems more usable (*e.g.,* test assessment)
5. To offer a modular architecture that allows multiple methods of proving

It is perhaps too early for empirical studies of classroom experience using the enhancements to GeoGebra. In this respect the situation of GeoGebra is similar, but not identical, to that of proof technology in general. While it is reasonable to expect proof technology to foster students' proving abilities, and there is certainly supporting anecdotal evidence, its potential advantages have not yet been systematically assessed.

Theorem provers in mathematics education: challenges

Automated proof assistants present challenges to mathematics educators because they were conceived as tools for research in computer science and mathematics, and not with teaching concerns in mind. Nevertheless, mathematics education could benefit from examining how they might be of benefit as teaching tools, in particular in the teaching of proof.

Confidence versus understanding

We know that automated proof assistants are designed to provide a guarantee of correctness, and indeed they are very good at establishing the validity of a proof. The question, then, is to what degree these tools can also be helpful in explaining *why* it is that a theorem is true. Important as it is to practicing mathematicians, this aspect of proof is the prime consideration for mathematics educators, who have always placed such high value on proofs that not only prove a theorem but also provide insight and explanation. (This topic has been explored extensively in Hanna, 2018, and in almost all the literature on proof in mathematics education).

There is a consensus, in fact, among mathematicians and mathematics educators alike that a proof has two potential roles, a guarantee and an explanation. Theorem provers do provide a guarantee, as we have seen, but in the shape of a fully formal proof that may be unintelligible. They were not designed to identify or highlight the main mathematical ideas behind a proof, and so it is no surprise that they fail in the explanatory role.

This state of affairs is a challenge for educators. They are aware that automated proof assistants are part of today's mathematical practice, and that it would be desirable that they be reflected in the mathematics curriculum for that reason alone. They also have reason to believe, based on the anecdotal evidence, that this new proof technology could turn out to be of great benefit in the classroom. On the other hand, their focus will continue to be fostering students' understanding of mathematical ideas, including that of proof, using the tools available to them.

In the longer term, increased use of automated proof assistants and further experience with them under research conditions may also provide software developers with both the incentive and the understanding of usability that they require if they are to incorporate features that would make them more suitable to pedagogical aims.

Reconsidering the role of logic in teaching proof with understanding

To learn proof with the help of automated proof assistants, one must have some knowledge of symbolic logic. This requirement is not specific to computer-assisted proving, of course, but simply reflects the role of symbolic logic in teaching proof in general. This role has been described by several mathematics education researchers who examined teaching proof from three different perspectives, psychological, educational, and contextual.

Durand-Guerrier, Boero, Douek, Epp and Tanguay (2012), for example, reviewed a large number of studies that weighed the epistemological and didactic arguments for incorporating the principles of formal logic into the undergraduate curriculum. In conclusion they said that "This brief review reinforced our position that it is important to view logic as dealing with both syntactic and semantic aspects of the organization of mathematics discourse" (p. 385). While recognizing that familiarity with logical principles alone is not sufficient to understand a proof, they saw it as necessary, and thus recommended the explicit teaching of the rules of first-order logic. They point out that a mastery of elementary logic can help students "avoid invalid deductions, and comprehend the basic structures of both mathematical proof (direct and indirect) and disproof by counterexample" (p. 374). Their conclusion is that teaching logical principles is essential, but that for each specific proof it must be accompanied by a focus on the relevant mathematical subject matter, supported by appropriate examples and counterexamples.

Experience in learning proof with the help of a theorem prover does provide some evidence for the value added by the software in forcing an awareness of the logical structure of a proof. The following is a comment offered by a student who used the theorem prover Lean:

> I believe learning Lean has brought great clarity to my understanding of mathematical proofs. In a way it's like the perfect scratch pad, focusing your mind on the goal while keeping track of all your assumptions and checking your thinking. Maybe I would've had an easier time learning proofs if I'd used Lean in the first place, although I have no way of knowing. [10]

Conclusion

How best to teach proving so students can both construct proofs and understand the mathematics behind them—that remains the central challenge. Although mathematics educators have long dedicated time and effort to this issue, research continues to show that students have difficulty with proof, at both the secondary and the post-secondary levels. Computer-based proof technology may well offer a new option in overcoming this difficulty, while at the same time creating an opportunity to make mathematics education more reflective of modern mathematical practice.

Of course, these new tools cannot be introduced successfully without considerable thought and research on the part of educators, including a thorough analysis of classroom instruction with and without their use. Some questions quickly come to mind. Can automated proof assistants, based as they are on mathematical logic, help students clarify their thinking and improve their reasoning abilities? Can they enhance understanding by helping students focus on structuring arguments correctly? Can they help students move from a formalized proof to an informal proof that does highlight key mathematical ideas? Most broadly, just what is the most productive relationship between proving with a proof assistant and advancing mathematical understanding?

Educators will need to consider not only how existing tools can best be used, but also whether they require modification (or even replacement) to make them suitable for an educational setting. Balacheff and Boy de la Tour (2019) reviewed some of the literature on existing proof assistants and found them to be far from satisfactory. They went on to offer criteria that must be taken into account in making proof assistants more useful for teaching and learning.

> Along with providing ATP features for mathematicians, computer-based tutors must take three additional categories of users into account: the curriculum decision makers (who specify the standard of mathematical validation at a given grade), the teachers (who orchestrate learning and decide what counts as a proof in relation to a standard), and the learners (who are simultaneously constructing an understanding of proof and of the related content). (p. 356)

Proof assistants that meet the requirements of these stakeholders will never be developed in the absence of initiative on the part of mathematics educators and a demonstrated demand fuelled by increased use. Secondly, success also requires new and effective teaching strategies. These two efforts stand in a reciprocal relationship, so that the full benefit of proof assistants will be seen only over time as new teaching strategies effect the demand for new tool features and vice versa. The responsibility for both efforts rests squarely on the shoulders of educators. The key is to make a start, beginning with exploratory studies of the potential of these new tools at both the secondary and post-secondary levels.

Acknowledgements

We are grateful to the three anonymous referees for comments on drafts of this article. We wish to acknowledge the generous support of the Social Sciences and Humanities Research Council of Canada.

Notes

[1] In this article we will use the terms 'ATP', 'ITP', 'theorem provers' and 'proof assistants' interchangeably.

[2] Examples of proofs using automated tools can be found in Botana *et al.* (2015) (secondary school level), Avigad (2019), and Ganesalingam & Gowers (2017) (university level).

[3] One referee pointed out that "There is a wide body of work on 'intelligent tutoring systems' that have been applied to the teaching of proof. The idea is that these tutoring systems have 'expert algorithms' for proving theorems in some domain (usually geometry) as well as common models of students' thinking." We agree with this referee that "there is more out there than we are surveying." The focus of the present article, however, is on ATPs. Readers might want to follow the referee's advice by searching 'intelligent tutoring mathematics proof' and paying "attention to this broad body of research in computer science and cognitive science journals that mathematics educators continue to ignore to their detriment."

[4] Cited in Ornes, S. (2020, August). How close are computers to automating mathematical reasoning? Quanta magazine. https://www.quantamagazine.org/how-close-are-computers-to-automating-mathematical-reasoning-20200827/

[5] https://leanprover.github.io/about/

[6] Avigad, J., de Moura, L. &Kong, S. *Theorem Proving in Lean Release 3.23.0* https://leanprover.github.io/theorem_proving_in_lean/theorem_proving_in_lean.pdf

[7] Rorvig, M. (2019) Number theorist fears all published math is wrong. *Motherboard Magazine.* https://www.vice.com/en/article/8xwm54/number-theorist-fears-all-published-math-is wrong-actually

[8] Professor Kevin Buzzard: https://www.imperial.ac.uk/people/k.buzzard

[9] See https://leanprover-community.github.io/

[10] Andrew Helwer: https://ahelwer.ca/post/2020-04-05-lean-assignment/

References

Avigad, J. (2019) Learning logic and proof with an interactive theorem prover. In Hanna, G., Reid, D. & de Villiers, M. (Eds.) *Proof Technology in Mathematics Research and Teaching,* 277–290. Springer.

Avigad, J. & Harrison, J. (2014) Formally verified mathematics. *Communications of the ACM* **57**(4), 66–75.

Balacheff, N. & de la Tour, T.B. (2019) Proof technology and learning in mathematics: common issues and perspectives. In Hanna, G., Reid, D. & de Villiers, M. (Eds.) *Proof Technology in Mathematics Research and Teaching,* 349–365. Springer.

Botana, F., Hohenwarter, M., Janičić, P., Kovács, Z., Petrović, R., Recio, T. & Weitzhofer, S. (2015) Automated theorem proving in GeoGebra: current achievements. *Journal of Automated Reasoning* **55**(1), 39–59.

Bundy, A. (2011) Automated theorem provers: a practical tool for the working mathematician? *Annuals of Mathematics and Artificial Intelligence* **61**(1), 3–14.

Buzzard, K. (2020) Proving theorems with computers. *Notices of the American Mathematical Society* **67**(11), 1791–1799.

Durand-Guerrier, V., Boero, P., Douek, N., Epp, S. & Tanguay, D. (2012) Examining the role of logic in teaching proof. In Hanna, G. & de Villiers, M. (Eds.) *Proof and Proving in Mathematics Education: The 19th ICMI Study,* 369–389. Springer.

Ganesalingam, M. & Gowers, W.T. (2017) A fully automatic theorem prover with human-style output. *Journal of Automated Reasoning* **58**(2), 253–291.

Hales, T. (2008) Formal proof. *Notices of the American Mathematical Society* **55**(11), 1370–1380.

Hanna, G. (2018) Reflections on proof as explanation. In Stylianides, A.J. & Harel, G. (Eds.) *Advances in Mathematics Education Research on Proof and Proving,* 3–18. Springer.

Hanna, G., Reid, D. & de Villiers, M. (Eds.) (2019) *Proof Technology in Mathematics Research and Teaching.* Springer.

Harrison, J. (2008) Formal proof—theory and practice. *Notices of the American Mathematical Society* **55**(11), 1395–1406.

Hohenwarter, M., Kovács, Z. & Recio, T. (2019) Using automated reasoning tools to explore geometric statements and conjectures. In Hanna, G., Reid, D. & de Villiers, M. (Eds.) *Proof Technology in Mathematics Research and Teaching,* 215–236. Springer.

Hoyles, C. (2018) Transforming the mathematical practices of learners and teachers through digital technology. *Research in Mathematics Education* **20**(3), 209–228.

Melis, E. & Leron, U. (1999) A proof presentation suitable for teaching proofs. In Lajoie, S.P. & Vivet, M. (Eds.) *Proceedings of the 9th International Conference on Artificial Intelligence in Education,* 483–490, IOS.

Selden, A. (2012) Transitions and proof and proving at tertiary level. In Hanna, G. & de Villiers, M. (Eds.) *Proof and Proving in Mathematics Education: The 19th ICMI Study,* 391–422. Springer.

Voevodsky, V. (2014) The origins and motivations of univalent foundations. *The Institute Letter* (Summer 2014) 8–9.