Sociomathematical Norms and Automated Proof Checking in Mathematical Education: Reflections and Experiences

Merlin Carl

Europa-Universität Flensburg

Follow this and additional works at: https://scholarship.claremont.edu/jhm

Part of the Artificial Intelligence and Robotics Commons, Arts and Humanities Commons, Educational Technology Commons, Mathematics Commons, Science and Mathematics Education Commons, and the University Extension Commons

Recommended Citation


©2024 by the authors. This work is licensed under a Creative Commons License.

JHM is an open access bi-annual journal sponsored by the Claremont Center for the Mathematical Sciences and published by the Claremont Colleges Library | ISSN 2159-8118 | http://scholarship.claremont.edu/jhm/

The editorial staff of JHM works hard to make sure the scholarship disseminated in JHM is accurate and upholds professional ethical guidelines. However the views and opinions expressed in each published manuscript belong exclusively to the individual contributor(s). The publisher and the editors do not endorse or accept responsibility for them. See https://scholarship.claremont.edu/jhm/policies.html for more information.
Sociomathematical Norms and Automated Proof Checking in Mathematical Education: Reflections and Experiences

Cover Page Footnote
This work was presented at the (virtual) Formalize(-2) workshop in Zürich. We thank the organizers José Antonio Pérez Escobar and Deniz Sarikaya for the opportunity to present our work there and the incentive to lay down these considerations in written form. We thank six of our colleagues for providing the example solution texts evaluated in section 2.4. Moreover, we thank our two anonymous referees for several points that helped to improve both the content of the paper and its exposition.

This work is available in Journal of Humanistic Mathematics: https://scholarship.claremont.edu/jhm/vol14/iss2/4
Sociomathematical Norms and Automated Proof Checking in Mathematics Education: Reflections and Experiences

Merlin Carl

Europa-Universität Flensburg, Flensburg, GERMANY
merlin.carl@uni-flensburg.de

Abstract
According to a widely held view, mathematical proofs are essentially (indications of) formal derivations, and thus in principle mechanically checkable (this view is defended, for example, by Azzouni [3]). This should in particular hold for the kind of simple proof exercises typically given to students of mathematics learning to write proofs. If that is so, then automated proof checking should be an attractive option for math education at the undergraduate level. An opposing view would be that mathematical proofs are social objects and that what constitutes a mathematical proof can thus not be separated from the social context in which it arises. In particular, such “sociomathematical norms” play a role in teaching situations, see, for example, Stephan [37]. There thus seems to be a tension between the inherent social nature of “natural” mathematical proofs and attempts at their automated verification. In this paper, we explore this tension both theoretically and on the basis of our experiences with developing the Diproche system, a program for the automated verification of natural language proofs in college-level introduction to proof classes (modelled after the example of the Naproche system by Cramer, Koepke et al. [14]) and its use at the Europa-Universität Flensburg during the winter term 2020/2021.

1. Introduction
A rather natural way to look at mathematical proof texts is to consider them as proof texts, i.e., as justification of statements and thus as ways to establish their correctness with certainty. Regarded from this perspective, a mathematical proof is largely independent from its concrete linguistical shape.
The same proof may be expressed in many ways, in many languages, formalisms etc. In this view, proofs are ideal objects; the criterion for their correctness is their agreement with an absolute standard of logical rigor. This makes their correctness independent of historical, cultural, or social aspects. If the “absolute standard” in question can be laid down precisely, this makes the (strive for) mechanical verification of proofs a natural endeavour.

A different perspective would be to emphasize that proof texts are a means of human communication. As such, they are concrete, linguistic objects that appear in a certain historical, cultural, and social context. The criterion for their success is then the acceptance, after rational discourse, by the relevant community. As a consequence of this point of view, the correctness of proof texts is highly context-dependent, changes over history, and is socially constituted. Consequently, their verification requires the acquisition of and adherence to social norms.

These views are certainly complementary, but not necessarily contradictory. One may, for example, hold the view that proof texts are concrete linguistic objects, but that they are not themselves, but rather point to, “proofs”, which are ideal objects. Thus, for example, a vast amount of very different concrete linguistic objects has been named “Euclid’s proof for the infinity of primes”; but certainly, all their differences notwithstanding, there is a meaningful sense in which these represent the “same” proof. One may then well admit that the question whether this “pointing” works out depends on cultural, social, historical etc. factors, while retaining that there is an absolute correctness standard for “ideal” proofs. The dependency of such factors would then merely be a property of the “surface” of a proof, not the proof itself; and as long as we can still see, say, Euclid’s original text as a “source of inspiration” for constructing a proof text that meets the standards of our current context, the differences in standards can be regarded as “harmless” for the integrity of mathematics. Whatever the merits, and difficulties\textsuperscript{1}, of this approach might be is largely irrelevant for this work, which will be concerned with concrete proof texts.

\textsuperscript{1}What are these “ideal” proofs? How can we explain our access to them? Do we really have identity criteria for proofs? These are a few examples of the kind of annoying questions that can lead rather quickly into far murkier waters than one might initially expect.
Although they are not contradictory, there is still a certain tension between these perspectives. Often, this tension can be resolved by being precise about the sense in which one is using the term “proof”. There are, however, cases in which “proof” is taken simultaneously in both senses; and in such cases, the “tension” can lead to an actual conflict. One such case is the attempt to bring automated proof checking, with its reliance on absolute and stable norms of correctness, into mathematical practice, with its socially constituted norms for correctness and admissible communication. The particular instance of mathematical practice that we are concerned with in this paper is the correction of proving exercises in didactical situations.

Automated proof checking in mathematics has by now become a large and vivid field on the borderline of mathematics and computer science. Large-scale systems and libraries have been built up containing impressive amounts of machine-verified mathematical proofs, including highly advanced results like Hales’ proof of the Kepler conjecture (see, for example, [20]). In another direction, efforts have been made to allow such systems to process inputs that resemble natural (language) mathematical texts such as those encountered in textbooks or research articles. In recent times, these approaches have considerably advanced; see, for example, Frerix and Koepke [17].

The possibility of automated feedback on the correctness of mathematical proofs, and particularly proof texts written in (close to) natural language, suggests didactical applications: It is to be expected that providing students who are learning how to prove with an automated checking tool should be advantageous in several respects. And indeed, a number of systems in this spirit have been designed, such as Edukera,\(^2\) Lurch ([10]), QED-Tutrix ([28]), Elfe ([7]), Concludio or Diproche [6]. There are also a number of teaching projects for using automated proof checkers or proof assistants, e.g., in logic classes, such as [1] or [40]. Recently, Hanna and collaborators [21, 22] have directed attention to the didactical prospects of automated proof checking and proof assistants.

The Diproche (“Didactical Proof Checking”) system, described in detail in [6, 8], was designed by the author as a system in the spirit of Naproche (an automated proof checking system for proofs in natural mathematical English) specifically adapted to didactical applications. It accepts texts in a

\(^2\)See [https://edukera.com/](https://edukera.com/), last accessed on July 6, 2024.
controlled fragment of German and gives feedback on several levels about the correctness of the input. The system was employed during the winter term 2020/2021 in a first-year course at the Europa-University Flensburg with about 230 participants; an account of this, together with an evaluation, can be found in [9]. As expected, this large-scale real-life application revealed several interesting subtleties, in particular concerning tensions between the assumptions implicit in the use of automated proof checkers and various communicational, social, cognitive, and psychological aspects of actual (educational) mathematical practice.

At the surface, an automated proof checker implements a function that sends strings (e.g., texts in a controlled natural language) to the set \{“correct”, “incorrect”\}.

3 Every actual proof-checking software we are aware of gives feedback on single proof steps or lines rather than whole texts; however, the simplified picture we give is sufficient for illustrating our point.

The goal is to set up the system in such a way that its evaluations agree with the actual correctness, or otherwise, of the input. An implicit assumption in the background of the construction of automated proof checkers is then that the function to be implemented exists, i.e., that there is a uniform, precise, and stable standard of correctness for proof texts.

At first sight, this assumption may seem innocuous. Indeed, the absolute and rigorous standards for mathematical proofs have been pointed out over and over again, and occasionally taken as an ideal for other disciplines, seen for example in Spinoza’s attempt to build up ethics “more geometrico”. In [3], the philosopher Jody Azzouni speaks of the “benign fixation of mathematical practice”, an aspect which, according to him, singles out mathematics as a social practice.

However, this picture of mathematics as a stable social practice has been called into question by empirical research. In a study by Inglis et al. [25], a mathematical proof was given to 109 expert mathematicians for evaluation. Of these, about 73% considered it to be correct, while about 27% came to the conclusion that it had substantial gaps or flaws [page 276]. Only one study participant of the former group changed her/his evaluation after hearing the arguments of the latter group [page 279]. In general, the study participants did not even expect an “overwhelming” [page 278] majority to agree with their assessment. From this, Inglis et al. draw the conclusion that there
are “two different sources of disagreements that mathematicians may have when evaluating a proof: performance errors in the validation of a proof, and different standards about what constitutes a proof in an established branch of mathematics” [page 271].

If these two sources were the only reasons for disagreement about proof correctness, one should expect agreement to increase in strongly controlled contexts, such as expert mathematicians assessing solutions to beginner’s proof exercises. However, a study by Moore et al. [32] indicates that this is not the case. Given solutions to the exercise problem of proving that “$y - x \in \mathbb{Z}$” defines an equivalence relation on $\mathbb{R}$, the correctness scores assigned to the same solution by four professors of mathematics varied between 5 and 9.8 on a $1-10$ scale [page 257]. One reason identified in [32] for this phenomenon was that the scoring process was based on guesses about the student’s thoughts rather than a purely text-immanent approach, so that the same text passage could once be classified as a simple “slip” or as revealing a fundamental “misunderstanding” [pages 257 and 260].

These studies indicate that the correctness standards employed in the evaluation of mathematical proof texts, even in very controlled situations, are much more subtle compared to the “background assumption” of automated proof checking described above. Indeed, in this paper, we want to propose the thesis that, in didactical settings such as lecture courses, these standards are highly context-dependent and continually changing norms are constituted through negotiations in the social context of the lecture. In the literature, such socially constituted norms that are specific to aspects of mathematics have been called “sociomathematical norms”.  

If this thesis is true at least to some degree, it clearly represents an obstacle to using automated proof checkers for the purpose of teaching students how to prove.  

This obstacle poses a challenge that systems such as Diproche will have to face in at least two ways: Once technically, by setting up the system in such a way that norm changes can be accommodated, and once with respect to the integration of the system into teaching.

---

4 The concept, along with the term, was originated by Yackel and Cobb in their paper “Sociomathematical Norms, Argumentation, and Autonomy in Mathematics” [43].

5 Obviously, we are excluding teaching contexts in which proof formalization, automated theorem proving, or the use of proof assistants is (part of) the learning objective.
Let us point out here that the didactical goal of “learning how to prove” has several aspects, including, but (by far) not limited to:

1. Learning heuristic strategies, mathematical creativity, and “great ideas” (in the sense of Engel [15, page v]) for problem solving, such as using analogy and generalization, or looking for invariants, which is found, for example, in the works of Polya [34, 35] or in training books for mathematical competitions such as [15].

2. Learning the basics of deductive argumentation and structuring of arguments and how to properly present mathematical arguments in writing, which is often done in “introduction to proof” classes at the beginning of university education; this is, for example, the goal of books like Chartrand et al. [11].

3. Learning how to construct formal derivations in a given deductive system, perhaps using automated theorem provers, proof assistants or proof checking softwares such as Mizar, Coq or LEAN (see, for example, [1, 2, 21]).

There is little doubt that the use of automated proof checking is helpful in (3), while the imaginative and inventive constructions and arguments typically associated with (1) are currently beyond the scope of automated checking. It is (2) that we focus on in this paper, although our points concerning the importance of social norms and their negotiation applies to (1) as well.

1.1. Related Work

There are various automated proof checking systems for didactical purposes. Some are adaptable to arbitrary purposes, such as Lurch [10]; others are specific to certain topics, such as QED-Tutrix [28], which is designed for proofs in elementary geometry, or Terence Tao’s QED [39], which introduces the student to propositional calculus. Moreover, there are various didactical projects for using professional formalization tools, proof checkers, and proof assistants in mathematical education; see, for example, Avigad et al. [1, 2].

A lot of work has also been done on (close to) natural input languages for automated proof checkers; well-known examples include Naproche [13, 14] and SAD [41], which have recently been combined in the Naproche-SAD.

\(^6\)We thank one of our anonymous referees for pointing this out.
system [17, 18]. To the best of our knowledge, however, the only system other than Diproche that works with a controlled natural language is Elfe [7], which uses a much more restricted language and inference than Diproche.

The social aspects of mathematics in general, and of proof in particular, are addressed, for example, in Heintz [23] or de Millo et al. [31]. The relevance of the social character of mathematical proofs, and sociomathematical norms, for the concept of formal proofs and its use in educational settings has been discussed in CadwalladerOlsker [5]. A part of the considerable amount of literature on the role of sociomathematical norms in mathematical education will be discussed in the next section. So far, the relevance of sociomathematical proof correctness norms for automated proof checking in educational contexts seems to have received comparably little attention.

2. Sociomathematical Norms

As discussed above, automated proof checking relies on the assumption that there is a precise, stable, and uniform correctness norm for proof texts. This assumption would be doubtful in contexts in which such norms have social aspects to them, and are thus inherently fuzzy and subject to development and negotiation. And indeed, at least in didactical settings, such aspects are easily identified.

For the general area of mathematical education, Yackel and Cobb, in their paper “Sociomathematical Norms, Argumentation, and Autonomy in Mathematics” [43], introduced the term “sociomathematical norm” for such aspects of mathematics-related norms, where it is in particular applied to criteria for correct mathematical justifications at the elementary school level. The term is nicely defined in Stephan [37]:

“Sociomathematical norms are the normative criteria by which students within classroom communities create and justify their mathematical work. Examples include negotiating the criteria for what counts as a different, efficient, or sophisticated mathematical solution and the criteria for what counts as an acceptable mathematical explanation.” [37, page 1]

Most empirical investigations of sociomathematical norms pertain to elementary school education [43, 26]. Sociomathematical norms in university settings have been considered, for example, in Yackel, Rasmussen and King [44],
Stylianou [38], and Karadinata, Sugilar, Farlina and Kurahman [27]. Güven and Dede [19] contains a comparison of sociomathematical “classroom micro-cultures” in an advanced mathematical setting, considering subtle normative differences between different classrooms. The relevance of sociomathematical norms in mathematical practice outside of teaching situations was considered by Piatek-Jimenez [33]. To the best of our knowledge, sociomathematical norms for proof text correctness have so far not been considered in detail in the literature.

2.1. Higher math education as a normative interference field

In the last paragraph, we suggested that a number of sociomathematical normative decisions are in the background when evaluating a proof text in a didactical context.7 In this section, we want to suggest that, in academic teaching, students are confronted with at least two dimensions of normative changes:8 Once between different lectures or courses at the same time (such as linear algebra and analysis, two courses that are typically taken simultaneously by a beginning student of mathematics), which we will call “horizontal norm differences” and once between different points of time within the same course, which we will call “vertical norm differences”.9

Vertical norm differences appear to be an inevitable part of teaching mathematics: Techniques that are carefully introduced and applied very explicitly the first few times become routine later on, with more and more details being omitted10 and results obtained frequently becoming part of the methodol-
Thus, along a lecture, norms may become more liberal as a consequence of learning progress. It is also conceivable, although certainly rarer, that the reverse process takes place and steps allowed so far are suspended for a certain period. This could happen, for example, when an intuitively accessible theory like number theory, geometry, or the theory of the real numbers is first taken for granted and later on axiomatized, with exercises now asking students to derive very simple facts from the axioms that were previously taken for granted. A typical example would be to ask students to derive basic rules for calculating with fractions from the group laws for addition and multiplication, along with the law of distributivity.

Horizontal norm differences can occur, for example, when different lectures treat the same subject, or use the same concepts, at the same time from different perspectives. A typical example would be an analysis lecture running parallel with a linear algebra lecture, with the analysis lecture taking great pains to introduce the real numbers and prove their basic properties, while the linear algebra lecture takes finite-dimensional real vector spaces as their paradigmatic examples for vector spaces from the very beginning.

The norms by which a text is evaluated as an acceptable solution, or otherwise, of a given problem are thus highly context-dependent. They depend both on the overall context of the course in which the problem is posed, and on the time at which it is posed.

In mathematics, it is quite common that methods are framed as theorems. Thus, for example, König’s lemma or Zorn’s lemma are usually taught as theorems, but often used as proof strategies that heuristically guide the search for proofs.

So far, there seems to be little research on the question how sociomathematical norms differ between simultaneous lectures, how they develop within the same lecture, and what difficulties students may experience due to norm differences. We regard this as a promising field for empirical research.

These two dimensions are by no means comprehensive. For example, if an exercise asks for a proof of a part of a result that was given in the lecture, but with the respective part of the proof omitted and deferred to the exercises, then it is usually implicitly assumed, but rarely stated explicitly, that one is not allowed to use either this theorem or subsequent developments depending on it in the solution. This, of course, relies on the student’s having a sense for the “point of the exercise”. A somewhat amusing example of what can go wrong here is provided by students who answered an exam question asking them to prove that any two bases of a finite vector space have the same cardinality by stating that both cardinalities need to agree with the dimension of said space. There are certainly
2.2. Negotiation of sociomathematical norms in academical settings

Let us consider a usual setting for an introductory university lecture on mathematics. Such a lecture has typically well over a hundred students and is taught by a lecturer; in addition to the lectures, there are exercise classes or tutorials in which exercises and solutions are discussed with a tutor.

In such a setting, the negotiations of correctness norms for proof texts take place between a number of parties, the most obvious ones being the students, the lecturer, and the tutors. Less obvious parties are also involved: Other lecturers from the same department want students to be educated in such a way that they will be able to attend their more advanced lectures. If there is an official syllabus, or if the study program has to be accredited by the state, there are political constraints on such negotiations. Additional constraints come from the wider mathematical community by which the students should eventually be accepted etc. But let us focus on the three groups mentioned, for which the negotiations take place in the most explicit and obvious way.

Norm negotiations among the lecturer and the tutors are particularly clearly visible in the case of cooperative exam corrections. Typically in such settings, the same exercise is corrected by several people, and in order to ensure a consistent evaluation, norms are fixed in the form of a grading scheme. Usually, it is hard to come up with such a scheme a priori that is applicable to the solutions actually received. Hence, this is usually done by first looking through a number of solutions, identifying typical approaches and mistakes, and then discussing how strongly certain aspects and mistakes should be weighed. That different standpoints can occur in such discussions and that such explicit norm negotiations are necessary for ensuring a consistent grading can be seen, for example, from the study by Moore et al. [32] mentioned in the introduction.

contexts in which this argument makes sense; had someone who attended the course forgotten about this statement, reminding that person of the concept of dimension might well suffice to generate “true and justified belief”. From a logical perspective, however, the very definition of the concept of dimension relies on the statement in question, and in this respect, the argument is circular. This logical perspective, which requires students to pretend that they do not know the statement that they learned from their lecturer and tutors, read up in textbooks etc. to be true and need to justify it as if all of this had not taken place, may well appear to be quite unnatural to people with little experience in mathematics. For an account of how confusing logical questions can be for formally uneducated people, see, for example, Counihan [12].
Norm negotiations between lecturer/tutors and students usually take a more latent form. Feedback is provided for students’ solutions, in which normative decisions are implicitly expressed. Template solutions are, for example, given on the blackboard, which serve as examples. When solutions are presented by the tutors, students occasionally ask how explicitly certain steps need to be made or how much detail will be required in the exam. In such cases, the tutor may offer his assessment, thus setting up a norm.

2.3. Sociomathematical Norms for Proof Texts

In this section, we want to consider how sociomathematical norms, as introduced above, apply to proof texts. Here, we see at least three normative aspects of proof texts.

The first aspect concerns the fundamental question of what kind of argument can count as a mathematical proof in the first place. Although there seems to be a wide consensus on this point among research mathematicians, this is not necessarily the case for the general public or beginning students. For example, it frequently happens that a few examples are presented as a “proof” for a universal statement, or that the statement is merely reformulated in a different and more elaborate wording, or that arguments based on intuition (“I cannot imagine otherwise”, “It’s obvious”), language conventions (“This set is open, so it cannot be closed”), authority (“It’s in the book”, “The professor said so”), common consent (“Everyone knows that addition is commutative”) or familiar practice (“We could not talk about “the” dimension if there were two bases of a vector space with different sizes, but we do. Thus...”) are offered.

Once logical deduction has been established as the standard for a valid mathematical justification, the detailed standards for logical justifications yield a second normative aspect of proof texts. Here, the two obvious major points are what can be assumed as given and which inference steps are admissible.

---


15See, for example, Martin and Harel [30, page 42]: “The view point that a mathematical proof must be a deductive argument is certainly held by mathematically sophisticated persons. However, our experience suggests that persons with limited experience in mathematics (...) accept and provide examples as a legitimate process of mathematical proof.”

16Since lectures for introductory mathematics classes rarely start with a comprehensive
There are further aspects, such as whether one can use pictures or diagrams in a proof and if so, in what way, but these will not be considered here.)

Finally, the actual textual presentation of a logical argument is subject to another set of normative standards. Here, there are norms concerning text structure, acceptable formulations, notational conventions, use of formalism, the allowed degree of “abuse of notation” etc.

In order to be successful in constructing solutions to exercise problems, students need to acquire all of these norms. For beginning students, all of these three normative aspects can be challenging.\textsuperscript{17}

2.4. Micronorms in evaluating exercises: A case study

To illustrate the discussion above, we now consider normative aspects of a particular proving exercise. In particular, we consider the following problem from Chartrand et al. [11, page 90]:

Show that, if \( n \in \mathbb{Z} \), then \( n^2 + 3n + 5 \) is an odd integer.

A typical solution for this exercise might look as follows (text portions that we will discuss below are marked with red bold numbers):

\begin{proof}
Case 1: \{3\}
Suppose that \( n \) is even.
Then there is an integer \( m \) such that \( n = 2m \). \{4\}
Let \( m \) be an integer such that \( n = 2m \). \{5\}
Then \( n^2 + 3n + 5 = (2m)^2 + 2(3m) + 5 = 4m^2 + 6m + 5 = 2(2m^2 + 3m + 2) + 1 \). \{6\}
Hence \( n^2 + 3n + 5 \) is odd. \{7\} \text{qed.} \{8\}
Case 2: \ldots \text{qed.}
Hence \( n^2 + 3n + 5 \) is odd. \{9\} \text{qed.}
\end{proof}

As innocuous as this example may seem, there is already a plethora of subtle normative decisions that a corrector of a solution for this exercise needs to

\textsuperscript{17}This point is also discussed in CadwalladerOlsker [5, page 51].
make, and these decisions are rooted in the specific context in which the solution is put forward and the didactical methodology and goals that one pursues. Below, we classify several solutions to the exercise above according to their adherence, or otherwise, to the specified criteria.\footnote{We are not claiming that the our list is by any means exhaustive.}

\{1\} Should the variable \( n \) be introduced at the beginning of the text? Not introducing variables before using them is a very common mistake among beginning students. On the other hand, one might read the problem statement “If \( n \in \mathbb{Z} \)” as implicitly introducing \( n \) and declaring it as an integer, so that there is no need to repeat the introduction. In this view, one might even argue that this repetition is a formal mistake, as the variable \( n \) is now declared twice. In the classification, we write 1 when \( n \) is introduced as an integer, and 0 otherwise.

\{2\} Should one mention that, as an integer, \( n \) must be either even or odd, so that the case distinction is complete? Didactically, this might help to raise awareness on the student’s part for the structure of a case distinction, in which justifying its completeness is required in general. On the other hand, one might regard the completeness as so obvious in this case that insistence on mentioning this might be regarded as a pointless formality, with the potential negative effect that students learn to regard justifying the completeness of case distinction as pedantic in general. For this reason, this should at most (if at all) be required for the very first exercises concerning case distinctions, to be dropped soon afterwards. We will write 1 when some indication to \( n \) being either even or odd is given, and 0 otherwise.

\{3\} Is such a marker for a case distinction required? Requiring this forces students to write up their solution in a very structured way, thereby (hopefully) becoming more aware of the structure of their argument. In any case, the presentation must in some way make it clear that the assumption that \( n \) is even is a local assumption that is dropped when the next case is considered. (Otherwise, one would from that point on argue under the contradictory assumptions that \( n \) is both even and odd, which is certainly not what one wants). Hence, some structural markers should certainly be required. We write 1 when cases are started with explicit markers, and 0 otherwise.
Do we require the representations of even/odd numbers as $2k$ or $2k - 1$ and the corresponding term manipulations, or can we directly apply arithmetical background knowledge for odd and even numbers, such as “the sum of two odd numbers is even” or “the product of an odd and an even number is even”? This has a considerable influence on the admissible inferences. Whether or not this should be allowed depends on the didactical goals that one wants to achieve with the exercise. For example, if the point is to teach the heuristical move of going back to the definitions of the occurring terms, and perhaps introducing variables as representing examples for existential statements, then one should insist on introducing the representations. If, on the other hand, the goal is an exercise in arithmetic modulo 2, the other way is preferable. (The solutions to this and several related problems in Chartrand et al. [11] are all of the former kind.) We write 1 when the solution proceeds via substitutions, and 0 otherwise.

In the text above, the steps of stating the existence of an integer $m$ with $n = 2m$ and introducing $m$ as a referent for such an integer have been separated. Quite often, one of these steps would be omitted, by either taking the existential quantifier in the first sentence to implicitly introduce a referent or by taking the second step to implicitly contain the first as a presupposition. However, for didactical purposes, especially at a beginner level, there are reasons to keep them separate. First, it is logically a somewhat delicate business how the introduction of referents through existential quantifiers works: For example, the sentences “There is an integer $n$ such that $n = 0$. There is an integer $n$ such that $n = 1$.” should certainly not introduce $n$ as a referent for an integer that is both equal to 0 and to 1. On the other hand, leaving the existential statement out of the picture and simply writing “Let $n$ be...” hides the fact that this relies on the existence of such an $n$, which needs to be justified beforehand. Should this separation be part of the criterion for a correct solution? We write 1 when existential statement and the introduction of a referent are separated, and 0 otherwise.

In the Naproche system, where existential quantifiers are indeed processed as introducing referents, this is solved by the use of dynamic predicate logic; see Cramer [13]. As elegant as this solution is, it is certainly beyond the scope of what one would want to teach beginning students.

It is indeed another frequent mistake in beginning students’ solutions that “Let...”-sentences are used to conjure up objects from nowhere, including objects with contradictory properties.
How fine-grained do the term manipulations have to be? Would it be acceptable to simply write \( n^2 + 3n + 5 = 2(2m^2 + 3m + 2) + 1 \)? More specifically: Should substitutions—such as replacing \( n \) with \( 2m \)—be separated from other manipulations? Can multiplying out and gathering together terms be performed in one step or should this be at least two steps? Certainly, in an exercise like this, one should not require going all the way down to basic arithmetical laws like associativity, commutativity, and distributivity, lest the solution be blown up to lengthy shiftings of brackets, which diverts from the point of the exercise.\(^{21}\) In other contexts, however, precisely such a detailed manipulation may be required. We write 1 when some detail on the manipulation is given, and 0 otherwise.

For concluding that \( n^2 + 3n + 5 \) is odd from its representation in the form \( 2(2m^2 + 3m + 2) + 1 \), one actually requires that \( 2m^2 + 3m + 2 \) is an integer. Since \( m \) is an integer, one may take this to be “obvious”. On the other hand, it is an important point, and one may thus at least want to require that it is explicitly stated. It may help to avoid mistakes like “We have \( x = 2 \cdot \frac{7}{2} \), so \( x \) is even.”, which occasionally happen. Indeed, in the solutions in Chartrand \textit{et al.} [11], this is always done. Going back still further, and justifying that \( 2m^2 + 3m + 2 \) is indeed an integer because the integers are closed under addition and multiplication, seems to miss the point of the exercise, much like the detailed manipulation sequence in the last footnote. Should one regard a solution as incomplete when it is not mentioned that \( 2m^2 + 2m + 2 \) is an integer? We write 1 when it is made explicit that \( 2m^2 + 2m + 2 \) is an integer, and 0 otherwise.

How do we indicate that the case is finished? As in point \{3\}, this is an important marker for the logical structure of the text. We write 1 when there is an explicit end marker for cases, and 0 otherwise.

After both cases have been dealt with, the proof ends. However, one might argue that the final step of deducing the proof goal that \( n^2 + 3n + 5 \) is

\(^{21}\)To illustrate our point, a “complete” manipulation sequence for the equality above would require placing brackets into the additions in order to make the terms well-formed in the first case; and then, it would read something like this: \((n^2 + 3n) + 5 = ((2m)^2 + 3(2m)) + 5 = ((2m)(2m) + (3-2)m) + 5 = (((2m) \cdot 2) - m + (2-3)m) + 5 = (((((m \cdot 2) \cdot 2) - m) + (2-3)m) + 5 = (((((m \cdot 2) \cdot 2) - m) + (2-3)m) + 5 = (((((m \cdot 2) \cdot 2) - m) + (2-3)m) + 5 = (((((2 \cdot (2 \cdot m)) + 2 \cdot (3m) + (2 \cdot 4) + 1 = 2 \cdot ((2m^2 + 3m) + 2) + 1 This becomes much more cumbersome in the case that \( n \) is odd, which is left out in the text above.
odd from the treatment in the two cases, is missing and that this deduction
should be made explicit by a final statement such as “Thus \( n^2 + 3n + 5 \) is
odd.” after all cases are finished. This would in particular help to raise
awareness for the point that, in the cases, this statement is obtained under
certain extra assumptions, and that from this (and the completeness of the
case distinction, see above), we can now conclude that \( n^2 + 3n + 5 \) is odd
without any extra assumptions. On the other hand, one might also regard
this as a needlessly pedantic redundancy. We write 1 when the proof goal is
mentioned again after both cases have been dealt with, and 0 otherwise.

To get some idea of how similar solutions written by experts would be with
respect to these normative details, we asked six of our colleagues at the
math department of the Europa-Universität Flensburg to write solutions to
the problem above that they would accept as a solution from a student. We
then annotated them according to the scheme above, along with the solution
given in Chartrand et al. [11]. We also wrote two solutions that were accepted
by Diproche for comparison. The colleagues we asked had all been involved
in teaching a first-year course that used exercises of this kind for introducing
students to proving techniques, and had exchanged and presented solutions
for such exercises. There was thus a shared context for those who wrote the
solutions, so that one might expect an increased agreement in terms of norms.
Still, the solutions received were quite dissimilar when compared with respect
to the 9 points above. Table 1 summarizes the results, where “/” indicates
that the respective criterion was not applicable; for example, one solution
neither used a case distinction nor term manipulations.

Even a superficial glance at this—admittedly sparse—data set reveals a con-
siderable disparity between the solution texts with respect to adherence to
the normative decisions above.

This is confirmed by some quantitative considerations: None of the nine pro-
posed norms was adhered to by all solutions; the average norm was adhered
to by 2.44 (rounded to two decimal places) of the 7 expert solutions (including
Chartrand, but excluding the “artificial” solutions written by the author
specifically for the Diproche system).

\[^{22}\text{ Obviously, we do not claim that this is a cogent empirical survey. But it still serves}
\text{to illustrate and substantiate our point.}\]
Moreover, only one expert’s solution (that of colleague 6) adhered to more than half of the proposed normative decisions. On the other hand, two expert’s solutions (those of colleagues 2 and 3) violated more of these than they adhered to. It is a bit cumbersome, but not hard to check\textsuperscript{23} that the same picture arises for every non-empty subset of these nine proposed norms: For every such subset, at least one expert solution violates more norms in this subset than it adheres to (non-applicable norms were ignored in the evaluation).

Thus, any stance that one takes with respect to the 9 criteria presented would have the effect that some solutions written by experts who have clearly mastered the subject in question would receive a negative evaluation.

This reveals a difficulty with using automated proof checking in didactical settings. If an automated system is to evaluate, or comment on, the correctness of a mathematical proof text, then normative subtleties such as those mentioned above need to be decided one way or the other. But then, the solutions accepted by the system become highly specific, with the consequence that many valid solutions will be reported as flawed. One possible way out could be to make the system “robust” with respect to such details, for example, by making the deduction machinery in the background strong enough to complete even large deductive gaps; but then, without a handle on issues such as deductive granularity, the system could not fulfill its didactical purpose.

\textsuperscript{23}We used a small Python program for this purpose.
In the most unfortunate case (which, however, could well arise for exercises such as the above with modern automated theorem provers), the deductive powers of the system suffice to solve the whole exercise, so that the empty string would be accepted as a correct solution. Imposing restrictions on ways to approach problems is an essential part of teaching. (For a drastic example, consider how addition exercises in elementary school make little sense when calculators are allowed.) Normative aspects are thus essential in teaching, and a system that is to be employed in teaching needs to take these into account in some way.

The normative divergence discussed above may come as a surprise to those who, like the author, have experienced that scores assigned to the same solution by different experts when, e.g., grading mathematical exams often show a strong agreement, with deviations in scores rarely exceeding one or two points. One should keep in mind, though, that in such settings, a “grading scheme” has usually been decided on beforehand (often after some process of norm negotiation, treating aspects such as “should this really be worth ... points, given that this other aspect gives only ... points”), so that the relevant normative aspects are fixed and agreed upon and there is less room for normative variance.\footnote{For differences in expert evaluations of mathematical proofs when no prior negotiation on standards has taken place, recall again Moore [32] (for the didactical case) and Inglis et al. [25] (which concerns proof correctness in a non-educational setting).}

3. Diproche

Diproche (“Didactical Proof Checking”) is a system for automated proof checking of proof texts in a controlled fragment of German, specifically designed for beginning student proof exercises, thus concerning the second of the three interpretations of “learning how to prove” mentioned in the introduction. Users can enter a text in a text window; the system then provides feedback on the correct use of language, type mistakes (which would flag, e.g., the sentence “Assume $A$.” when $A$ is a set or an integer), logical correctness of inference steps, presumable applications of formal fallacious inference rules (which would, for example, propose that “$(a + b)^2 = a^2 + b^2$” is due to a misapplication of the binomial formula), the fulfillment of proof obligations (was the announced (sub-)goal actually achieved?) and, in some cases, coun-
terexamples to false proof steps. The project drew its inspiration and its rough architecture from the Naproche (“Natural language Proof Checking”) system (see, for example, [14]). An account of the Diproche system can be found in [6] or in [8]. The system was used in a lecture course on introductory proof-based mathematics with about 230 students during the winter term 2020/2021; an account of the use of the system and its evaluation can be found in [6].

For each of the feedback categories just mentioned, the system implements a set of corresponding (implicit) normative decisions. Most obviously, the use of a controlled natural language implies a linguistic normative standard, which regulates acceptable symbols, vocabulary and sentence structure. With few exceptions, Diproche follows a rather strict “one step per sentence”-policy, and a clear distinction between declarations, assumptions, claims and annotations (such as “QED” or “⇒” as an indication that one is now proving one half of an equivalence statement). As a seemingly trivial, yet non-negligible, aspect, this includes an easily machine-readable standardization of formulas that is at the same time easily usable for students with no background in writing mathematical texts in text processing systems. Unlike, say, LaTeX, the Diproche editor does not require the user to mark formal expressions in the text. It seems didactically advisable to avoid such a requirement, as, in particular for users who are inexperienced in programming, it is likely to lead to many unnecessary, puzzling, and frustrating error messages which divert from the actual goal of learning how to prove (for example when students forget to mark variables as such and thus receive a parsing error). This shifts the burden to identify formulas to the system.26

25Thus, for example, the following would not be accepted by Diproche: “If \( n \) is an even integer, then so is \( n^2 \). Hence \( n^2 + 1 \) is odd.” First, in Diproche, we need to separate the variable introduction “is an integer” from the assumption “is even”. Second, Diproche does not incorporate abbreviations such as “so is”, so one would have to write “then \( n^2 \) is even”. Third, Diproche reads the “If ... then ...”-sentence as a proposition, while in this example, it has two structural purposes, namely introducing the assumption that \( n \) is even and then drawing a conclusion from this assumption. For Diproche, however, the “if”-part is no longer available after the sentence is finished, and so the second sentence would be checked with no assumption on \( n \), so that the step would be rejected.

26In Diproche, this is solved by a rather easy rule: If a string contains either a mathematical symbol (such as +, &., ∪, or <) or consists of a single letter, it is processed as a mathematical expression. This rule works well in German, since single letters only occur as variables. When working on an English version of Diproche, however, one immediately
(As an example for a normative restriction arising out of this decision, the customary omission of the multiplication sign is not allowed, and on the other hand, as many symbols as possible are represented in a way that is available on a standard keyboard, so that \((a^2b^2)^3\) has to be written as \((a^2 \cdot 2 \ast b^2 \ast 2) \ast 3\).) Next, there is the **logical** normative dimension, which concerns acceptable inference steps. Finally, there are the **structural** norms; an example of such a norm would be the assumption that a proof obligation is only fulfilled when it is stated as the last sentence of a purported proof text of it, or that assumptions and variable declarations are only valid within the paragraph in which they occur.\(^{27}\)

### 3.1. Diproche and Norm Changes

In all three normative dimensions just discussed, horizontal and vertical norm differences can (and typically do) occur. New symbols, words, and formulations are introduced during a course, and, as the topic advances, linguistic standards on exposition may become somewhat more liberal. New kinds of inferences become available, and larger gaps in inferences become acceptable; and while new structural markers may be introduced, their explicit use may become less strictly required. As an example for the last point, when proofs by induction are introduced, it is typically required of the students to write them up in a very clearly structured way, announcing that an inductive proof is about to come and then using explicit indications of the base case and the inductive step and a clear separation between them. At some point, it often becomes admissible to give an argument for the inductive step and then stating that “by induction, it follows that...”, perhaps leaving out the base case altogether in case it is trivial.\(^{28}\) In order to keep the disruption through norm divergences between the lecture and the system as small as possible, the possibility of norm changes needs to be taken into account in the design of the system.

There are two obvious ways to accommodate norm changes in the use of a system like Diproche in teaching: technically, by designing the system in a

---

\(^{27}\)With the exception of the paragraph immediately after the start of a (sub-)proof. See [6] for details.

\(^{28}\)For a discussion of the explication of such mathematical “frames”, see [16].
certain way; and didactically, which pertains to ways in which the system is used in teaching.

Already when implementing Diproche, the phenomenon of vertical norm changes was anticipated. For this reason, the automated theorem prover (ATP) working in the background of Diproche can be fine-tuned according to so-called “degrees of difficulty”. Each inference rule in the ATP has a unique identifier, and a degree of difficulty is a set of inference rules. Since new degrees of difficulty can be defined at any time, rules can in principle be allowed or forbidden in a different way for each separate exercise in this way. In general, as learning proceeds and exercises get harder, the admissible ATP rules become more and more liberal. (Note that this means that solving the same exercise in a higher “degree of difficulty” will in general be easier, and the corresponding proof texts will in general be shorter.) Moreover, the ATP rules are designed to mirror natural deductive patterns that typically appear in (correct) solutions. Among these rules are basic logical and topic-specific rules (such as deducing the existence of an \(m\) with \(n = 2m\) from the assumption that \(n\) is even), but also meta-rules that, for example, allow one to take all parts of conjunctions in the set of available assumptions for granted rather than demanding that they be explicitly deduced prior to their use.

To illustrate the explanations above, we give here the two Diproche-solutions that were included in Table 1. The first operates at a very low degree of difficulty, so that existence claims and variable introductions need to be separated, and one needs to work directly with the definitions of “even” and “odd”. The second example is from a higher “degree of difficulty” in which one can operate with some arithmetical background knowledge of odd and even numbers rather freely. If entered in the context of the difficulty degree for which the first solution text was written, it would be rejected.

---

29 As a consequence, the ATP works with several topic-specific submodules, each of which consists of a few hundred rules on average. As a result, the approach taken by some systems to require the user to justify each inference step by explicitly mentioning the underlying inference rule becomes infeasible when proceeding beyond the most basic proof tasks if one wants to keep proofs “natural”.

30 Recall our discussion of “degrees of difficulty” in Diproche. We are not claiming that the second text is more difficult to read or to produce, but that more prover rules were allowed in the setting in which it is checked (which in fact makes its production easier).
Moreover, in this solution, explicit case markers are omitted, and there are sentences that comprise several inference steps.

Here is the detailed Diproche solution with basic difficulty degree:

\[
\text{Es sei } n \text{ eine ganze Zahl. Zeige: Dann ist } n^2 + 3 \cdot n + 5 \text{ ungerade.}
\]

**Beweis:**

Fall 1: Angenommen, \( n \) ist gerade.
Dann existiert eine ganze Zahl \( x \) so, dass \( n = 2 \cdot x \).
Es sei \( x \) eine ganze Zahl so, dass \( n = 2 \cdot x \).
Also ist \( n^2 + 3 \cdot n + 5 = (2 \cdot x)^2 + 3 \cdot (2 \cdot x) + 5 = 4 \cdot x^2 + 6 \cdot x + 5 = 2 \cdot (2 \cdot x^2 + 3 \cdot x + 2) + 1 \).
Also ist \( n^2 + 3 \cdot n + 5 \) ungerade. qed.

Fall 2: Angenommen, \( n \) ist ungerade.
Dann existiert eine ganze Zahl \( y \) so, dass \( n = 2 \cdot y + 1 \).
Es sei \( y \) eine ganze Zahl so, dass \( n = 2 \cdot y + 1 \).
Also ist \( n^2 + 3 \cdot n + 5 = (2 \cdot y + 1)^2 + 3 \cdot (2 \cdot y + 1) + 5 = 4 \cdot y^2 + 10 \cdot x + 9 = 2 \cdot (2 \cdot y^2 + 5 \cdot x + 4) + 1 \).
Also ist \( n^2 + 3 \cdot n + 5 \) ungerade. qed.

Damit ist \( n^2 + 3 \cdot n + 5 \) ungerade. qed.

Here is the English translation:

Let \( n \) be an integer. Show: Then \( n^2 + 3 \cdot n + 5 \) is odd.

**Proof:**

Case 1: Suppose that \( n \) is even. Then there is an integer \( x \) such that \( n = 2 \cdot x \). Pick an integer \( x \) such that \( n = 2 \cdot x \). Then we have \( n^2 + 3 \cdot n + 5 = (2 \cdot x)^2 + 3 \cdot (2 \cdot x) + 5 = 4 \cdot x^2 + 6 \cdot x + 5 = 2 \cdot (2 \cdot x^2 + 3 \cdot x + 2) + 1 \).
Hence \( n^2 + 3 \cdot n + 5 \) is odd. qed.

Case 2: Suppose that \( n \) is odd. Then there is an integer \( y \) such that \( n = 2 \cdot y + 1 \). Let \( y \) be an integer with \( n = 2 \cdot y + 1 \). Then \( n^2 + 3 \cdot n + 5 = (2 \cdot y + 1)^2 + 3 \cdot (2 \cdot y + 1) + 5 = 4 \cdot y^2 + 10 \cdot x + 9 = 2 \cdot (2 \cdot y^2 + 5 \cdot x + 4) + 1 \).
So \( n^2 + 3 \cdot n + 5 \) is odd. qed.

Thus \( n^2 + 3 \cdot n + 5 \) is odd. qed.
Here is the concise Diproche solution with advanced difficulty degree:

Es sei \( n \) eine ganze Zahl. Zeige: Dann ist \( n^2 + 3 \times n + 5 \) ungerade.

**Beweis:**

Angenommen, \( n \) ist ungerade. Dann sind \( n^2, 3 \times n \) und 5 ungerade. Also ist \( n^2 + 3 \times n \) gerade, und damit ist \( n^2 + 3 \times n + 5 \) ungerade.

Nehmen wir nun an, dass \( n \) gerade ist. Dann sind \( n^2 \) und \( 3 \times n \) gerade, also ist auch \( n^2 + 3 \times n \) gerade, und folglich ist \( n^2 + 3 \times n + 5 \) ungerade.

Also ist \( n^2 + 3 \times n + 5 \) ungerade. QED.

And here is the English translation of the proof:

**Proof:**

Suppose that \( n \) is odd. Then \( n^2, 3n \) and 5 are odd. Hence \( n^2 + 3n \) is even, and thus \( n^2 + 3n + 5 \) is odd.

Now suppose that \( n \) is even. Then \( n^2 \) and \( 3n \) are even, so that \( n^2 + 3n \) is even, and hence \( n^2 + 3n + 5 \) is odd.

Consequently, \( n^2 + 3n + 5 \) is odd. qed.

Moreover, the feedback of the Diproche system is formulated in such a way that it yields suggestions for improving the text rather than definitive statements on its correctness. Thus, rather than reporting a certain step as “wrong”, Diproche indicates that this step could not be verified. This is done to moderate the effects of normative differences to a certain extent. Finally, the system is designed in such a way that its components can be easily adapted, should the need arise. The developer, who was at the same time a tutor for the lecture in which Diproche was used, was collecting feedback on the system’s performance and making adaptations where this seemed advisable throughout the lecture. Some of these adaptations are discussed below in more detail.

### 3.2. Experiences

In this section, we describe some experiences gathered through using Diproche in teaching. This was done by having links to Diproche proof exercises on several weekly exercise sheets.
In several places, there arose discussions among the faculty, and occasionally also between the faculty and the students, whether certain formulations and inference steps—such as those concerning linguistic and the logical norms—should be allowed.

For example, one of the sheets contained the following exercise:  

**Let** $n$ be an integer. Show that, if $3 | (n - 1)$, then it follows that $3 | (n^2 - 1)$.  

A typical attempted solution text looks like this:  

**Proof**: Suppose $3$ divides $n - 1$. Pick an integer $i$ such that $3 \cdot i = n - 1$. Then we have $n = 3 \cdot i + 1$. Hence, we have $n^2 - 1 = (3 \cdot i + 1)^2 - 1 = 9 \cdot i^2 + 6 \cdot i + 1 - 1 = 9 \cdot i^2 + 6 \cdot i = 3 \cdot (3 \cdot i^2 + 2 \cdot i)$. Thus $3$ divides $n^2 - 1$. qed.

Initially, the system flagged the step “Then we have $n = 3 \cdot i + 1$.” as not verified. The reason was that such “implicit” term manipulations had to be made explicit. In order for the solution to pass through the checker, one had to insert sentence like “We have $(3 \cdot i = n - 1) \Leftrightarrow (n = 3 \cdot i + 1)$.” just before this sentence. After some discussion, this was regarded as too fine-grained. Thus, the question arose as to how to strengthen the system in such a way that “obvious steps such as this one” are accepted, but without making term manipulations obsolete altogether. Eventually, a rule was added that allowed linear manipulation steps without further justification, so that one could pass from $T_1 = T_2$ to $a \cdot T_1 + b = a \cdot T_2 + b$ for arbitrary real numbers $a$ and $b$, including equivalent variants of such terms. This is an example of how normative demands are reflected by changes to the system.

During the semester, a few such adaptations were made. Students noted this fact explicitly and positively in their evaluation of the system’s deployment. Keeping the system flexible and open to such changes is thus one way to account for normative decisions and changes.

---

31 German original translated to English by the author.

32 In the exercise above, for example, we would certainly not want the system to accept the equation $n^2 + 3 \cdot n + 5 = 2 \cdot (2 \cdot y^2 + 5 \cdot y + 4) + 1$ without any intermediate step.
An interesting observation, however, was that occasionally, such a liberalization of linguistic or logical standards triggered further expectations for formulations or inferences that the system should accept, leading into a potential difficulty: Changing the system in accordance with normative decisions has an “ad hoc”-character. If too many such changes accumulate, the system becomes obscure and unusable. In order to retain a coherent system, one thus needs to consider such adaptations rather carefully.

Another aspect was the normative interpretation of the system feedback by students. In the normative inference field concerning proof text correctness that students are confronted with—proofs presented in the lecture, by tutors, in textbooks, by other students, corrections, and markings received from tutors etc.—system feedback adds another normative standard that is in partial agreement and in partial collision with these other standards. It is thus to be expected—and it was observed—that students expect a distinction between “real” mistakes, those that concern the mathematical “substance”, and mere “system artifacts” that are due to the specific formal demands by the system. However, with the system feedback alone, these two types of mistakes are not easy to distinguish; it can then happen that what are actual logical mistakes that would also have been flagged by a human corrector are regarded as mere formalities, and thus as irrelevant.

A particularly illustrative instance of this was the following: Several students did not explicitly state the proof goal as the last statement of their proof text. In such cases, the system reported this goal as an unfulfilled proof obligation. As a consequence, such error messages were regarded as belonging to the sphere of “mere formalities”. In one case, then, a submitted solution had all occurring propositions reformulated as assumptions, so that the text consisted entirely of assumptions. Since assumptions do not generate a need for verification, the system found no logical flaws and only reported one unfulfilled proof obligation. For someone with the view that such error messages are irrelevant, this would appear to be an essentially correct solution. This example shows that, without a professional framing of the feedback from the proof checking system, the normative interpretation on the part of the students may lead them considerably astray.
4. Discussion

The social aspect of correctness norms for mathematical proof texts, along with the fact that these norms are subject to change and negotiation is a challenge for any attempt to use automated proof checking, which implements a set of fixed, stable and clear normative decisions, in mathematical lectures. However, one might regard this as a feature, rather than a bug, of automated proof checking: It sets an incentive for lecturers to set up and communicate such norms to the students, rather than expecting students to pick them up “along the way”. Indeed, one occasionally hears students express frustration about the lack of such explicit standards for proofs, complaining, for example, that “one can never be sure if it is enough or whether one needs more detail”.

Consequently, one might consider turning the tables around. Rather than attempting to integrate automated proof checking into traditional lectures, one could consider automated proof checking as the basic standard for normative decisions, and set up the lecture accordingly.\footnote{We are not aware of current work in education championing such an approach. In this sense, this discussion is concerned with a hypothetical position that one might naturally suggest at this point. There are, however, concrete indications that such a practice may be forthcoming, for there are repeated proposals to use automated checkability as a correctness criterion for mathematical proofs. Most prominently, we find this in the “QED manifesto”\cite{QED}: “The standard, impartial answer to the question “Has it been proved?” could become “Has it been checked by the QED system?”. Such a mechanical proof checker could provide answers immune to pressures of emotion, fashion, and politics.” Although this passage does not specifically refer to educational settings, educational uses are mentioned as the third out of nine “motivations” for the QED project (which aims to build a system for the automated verification of mathematics, see for example \url{https://mizar.uwb.edu.pl/qed/}), and it is announced that “with increasing technology available, governments will look not only to cut costs of education but will increasingly turn to make education and its delivery more cost-effective and beneficial for the state and the individual.” One can easily imagine that automated checking will be more cost-effective, for instance, than hiring human tutors for correction purposes. In our view, it is thus worthwhile, and in fact crucial at this point, to point out why this would be a bad idea.} Thus, students would at the very start learn definitive standards for correctly expressed proof texts, which then would have to be strictly obeyed, both in the exercises and, preferably, also in the lecture itself. Setting aside the practical difficulties and the enormous effort that such an undertaking would bring with it, we believe that such an approach is in principle misguided for at least three reasons, which we list below in increasing order of importance.
First, as we already argued above, norm negotiations and changes allow for a didactically valuable flexibility in teaching. There is little point in insisting on painful formal details when someone has understood the point and could easily fill in the details when provided with an appropriate sketch. Indeed, strictly enforcing formal norms may be just as frustrating, or even more so, than a certain degree of uncertainty about norms.

Second, at least as long as the whole of mathematics is not based on automated formal methods, that is, as long as mathematical arguments are written up for human readers, norm acquisition is an integral part of mathematical education. Learning the standards for a good presentation of a valid argument from examples and adapting to the development and context-dependent variation of such standards are thus abilities that should be promoted and challenged in mathematical education.\textsuperscript{34}

Third, and most importantly, we believe that such a move would alienate students from the nature of proofs as a means for generating rationally justified conviction. One can learn to decide whether a given text agrees with a given set of normative criteria, and one may even learn to construct texts that meet such criteria, while still missing the whole point of a proof, and failing to see the need for proof. One may, for example, learn to construct inductive proofs “mechanically”, by just following the steps of the induction scheme, without grasping why these steps should convince one of the truth of the respective universal statement.

Mathematics is an area—perhaps even the exemplary area—where rational thinking by itself can solve problems and decide questions, with no reliance on authority; indeed, that is a crucial part of its educational value.\textsuperscript{35} Therefore, it would be misleading to present the verdict of a piece of software, no matter how well built, as the primary criterion for proof correctness. Rather, the teaching of proof should start with everyday examples of the ability of rational argumentation and deduction, and present mathematical argument as an extremely sophisticated development of these. Consequently, it should be made clear to the students why the steps in an argument are needed, and

\textsuperscript{34}This applies in particular to teacher education.

\textsuperscript{35}This point is made forcefully by Wittenberg, see [42, Chapter 1]. For the “democratic” character of mathematics, in contrast to the “authoritarian” way in which it is frequently taught, also see Lowsky [29].
why the seemingly artificial and contrived language in which the argument is expressed is an appropriate and natural means to do so, using relevant criteria such as comprehensibility, surveyability, and the generation of rational conviction. To achieve this, the student must be able to see how the correctness standards for proofs come out of the goals of generating rational conviction and communicating it to others. But then, the student’s successful participation in the negotiation of such norms is a genuine part of learning proofs. If norms are fixed and unchangeable because they are implicitly hard-wired in a piece of software, such participation is blocked.

We state this point as a thesis, which we believe to be a worthwhile subject for empirical investigation:

**Thesis:** The negotiation of standards of justification plays an important role in understanding and accepting these standards as rational criteria for the generation of justified belief, rather than externally superimposed norms.

If this is true, then it would be a strong reason to oppose attempts to dispense with norm changes and negotiations in favor of a set of norms imposed by the use of proof checking technology.

The questions of Hanna and Yan [22] remain of whether automated proof checking can and should play a role in teaching students how to prove and if so, how automated proof checking can be integrated into mathematical teaching in a productive way. We hope that this paper has helped to raise awareness for a challenge that such attempts will have to face.

---

36 This point can be seen as a transfer of the category of “intellectual autonomy” in mathematics discussed by Yackel and Cobbs ([43], p. 473f) in an elementary school setting to the contexts of proof correctness norms. See, e.g., Yackel and Cobbs, [43], p. 473-474: “(...) it is precisely because children can make personal judgements of this kind on the basis of their mathematical beliefs and values that they can participate as increasingly autonomous members of an inquiry mathematics community”

37 Some empirical evidence in favor of this thesis is given by the observation of what is called the “ritualistic” aspect of proof in the educational literature (see, for example, Martin and Harel [30], where the following observation was reported “(...) that students judged a mathematical proof on its appearance, relying on ritualistic aspects of proof.” [page 41], i.e., that a certain portion of students evaluates the question whether a given text represents a correct mathematical proof on the basis of entirely formal criteria.
On the other hand, the benefits of using automated proof checking, such as allowing immediate feedback on as many versions of a proof text as the student wants to try, seem to warrant continued efforts in this direction. One question to consider should then be how to keep the norms according to which the checking is performed transparent, flexible, and easily modifiable, so that the results of norm negotiations can be easily and quickly incorporated.\footnote{Compare with the discussion in Section 3.2.}

5. Acknowledgements

We thank our six colleagues for providing the solution texts evaluated in Section 2.4. Moreover, we thank the two anonymous referees for several points that helped improve both the content of the paper and its exposition. This work was presented at the (virtual) Formalize!\footnote{Compare with the discussion in Section 3.2.}-2 workshop in Zürich. We thank the organizers José Antonio Pérez Escobar and Deniz Sarikaya for the opportunity to present our work there and the incentive to lay down these considerations in written form.

References


56  Sociomathematical Norms and Automated Proof Checking


