



# Learning about Proof with the Theorem Prover LEAN: the Abundant Numbers Task

Athina Thoma<sup>1,3</sup>  · Paola Iannone<sup>2</sup>

Accepted: 24 March 2021 / Published online: 07 July 2021  
© Crown 2021

## Abstract

This exploratory study reports on characteristics of proof production and proof writing observed in the work of first-year university students who took part in workshops on the theorem prover LEAN (<https://leanprover.github.io>). These workshops were voluntary and offered alongside a transition to proof module in a UK university. Through qualitative analysis of 36 student produced proofs of an unfamiliar statement we highlight characteristics of proofs produced by students who did engage and who did not engage with LEAN. The analysis shows two characteristics of proofs written by students who engaged with the programming language. The first concerns proof writing and includes the accurate and correct use of mathematics language and symbols, together with the use of complete sentences and punctuations in proofs. The second concerns proof structure and includes the overt break down of proofs in goals and sub-goals. We conclude by hypothesising a link between the characteristics observed and the experience of engaging with the theorem prover and we reflect on the potential that engagement with this theorem prover may have in mathematics instruction at university level.

**Keywords** Proof · Automated theorem prover · Undergraduate mathematics

## Introduction

Proof is a central part of mathematics: being able to accept the values that this activity conveys is considered to be one of the characteristics of successful

---

✉ Athina Thoma  
a.thoma@uea.ac.uk

Paola Iannone  
p.iannone@lboro.ac.uk

<sup>1</sup> Department of Mathematics, Imperial College London, London, UK

<sup>2</sup> Mathematics Education Centre, Loughborough University, Loughborough, UK

<sup>3</sup> School of Education and Lifelong Learning, University of East Anglia, Norwich, UK

undergraduate mathematics students and an important step on the way to enculturation in the practices of mathematics (Dawkins & Weber, 2017). Therefore, investigations of university students' proficiency in proof writing, proof comprehension and how the teaching of proof impacts on those abilities form a big part of mathematics education research at university level. The overwhelming evidence from such research is that students struggle with many aspects connected to this activity (Harel & Sowder, 1998; 2007; Dreyfus, 1999; Weber & Alcock, 2004; Selden, 2011), often despite the good will and well thought through pedagogic intentions of their teachers (Weber, 2004). For this reason, several teaching interventions have been trialled to assist students with mathematics learning such as flipped classrooms (Talbert, 2015; Lo et al., 2017) and the introduction of enquiry-based courses in mathematics (Rasmussen & Kwon, 2007). Among those interventions, the inclusion of suitable programming mathematical languages in the curriculum continues to attract researchers' attention (Broley et al., 2018; Buteau et al., 2020). This paper reports on an exploratory study investigating the proof production and proof writing outcomes of first-year undergraduate students who voluntarily chose to engage with an automated theorem prover.

## **Programming Languages, Automated Theorem Provers and the Learning of University Mathematics**

In a recent survey of the use of programming languages in UK universities, Sangwin and O'Toole (2017) found that although the use of programming languages such as Maple, Python or R is becoming more common, there was no evidence of the use of automated theorem provers, even if some of them (e.g., Coq <https://coq.inria.fr>) have been used in mathematics research since the early nineties. The authors also report that the use of programming was mostly found in applied mathematics teaching (e.g., numerical analysis) and statistics, and that very seldom was linked to pure mathematics teaching. Yet very recent reports of the use of such languages suggest a positive impact on students' understanding of the necessity of mathematics rigour and subsequent advantages for proof production and proof writing (Avigad, 2019). In what follows, we review the literature on the use of programming languages in teaching undergraduate mathematics focusing on pure mathematics.

### **Use of Programming Software to Support Learning Mathematics**

Integrating the use of programming languages in mathematics research is now commonplace, while their use in mathematics teaching is much less so. In a recent survey (Broley, 2016) reports that 43% of the Canadian mathematicians surveyed for the study used computer programming in their research while only 18% included computer programming in their teaching. The author therefore notices a persistent disconnect between the practice of mathematics and its teaching. However, research in mathematics education has often suggested some benefits of using programming languages for the learning of pure mathematics.

Broadly speaking, two distinct types of programming languages have been used in university mathematics teaching: languages designed explicitly for teaching purposes, where their design has been informed directly by findings in mathematics education, and languages designed to advance mathematics research. Amongst the latter, of interest is the use of automated theorem provers for a branch of logic dealing with proving theorems by using computer programs both for checking existing proofs and (eventually) for writing new proofs. In the first category are programming languages such as Pandora (Broda et al., 2007), Sequent Calculus Trainer (Ehle et al., 2018) and ISETL (Dubinsky, 1995). In the latter are programming languages such as LEAN (<https://leanprover.github.io>) and COQ (<https://coq.inria.fr>)<sup>1</sup>. ISETL (<https://www.swmath.org/software/1370>) is probably the most popular programming language designed specifically to aid students with learning topics such as group theory, abstract algebra and combinatorics. ISETL was designed following the principles of APOS Theory (Dubinsky, 1984) and its impact on student learning has been widely documented (Dubinsky, 1995). Interesting to note here is that despite the wealth of material produced to support the use of ISETL (Dubinsky & Leron, 1994) and the positive outcomes that its use has for the learning of some pure mathematics topics (Pesonen & Malvela, 2000), this programming language is not used widely, and it is not in use in UK universities (Sangwin & O'Toole, 2017).

LEAN (<https://leanprover.github.io/>) is one example of a programming language designed for research in mathematics which has also been used for teaching undergraduate mathematics students (Avigad, 2019). The aim of LEAN, an open source theorem prover, is to bring interactive and automated reasoning together and build an interactive theorem prover with powerful automation and an automated reasoning tool that can check and produce (detailed) proofs (although these aims are not fully realised yet). LEAN is built on a verified mathematical library. It is a programming environment in which it is possible to compute with objects with precise formal semantics, reason about the results of computations, and write proof-producing automation. The LEAN project started in 2013 and currently both a downloadable and an online version of the software are available. LEAN's syntax has a close connection with the mathematical notation system, the properties and the mathematical objects can be supported by the software and the data types can be part of mathematical expressions. The development of this programming language, unlike that of ISETL, is driven by research in mathematics and more precisely by those in mathematics and logic who wish to construct a tool to check existing proofs and eventually produce new ones.

The online interface of LEAN is shown in Figs. 1 and 2. On the left hand side of the screen, the user writes the code, using the appropriate tactics (i.e., the programming moves which are allowed by LEAN), which is then processed by LEAN. On the right hand side of the screen, LEAN illustrates the changes in the goals of the proof,

---

<sup>1</sup>As an aside we note that Python, which is becoming a popular programming language in university mathematics according to Sangwin and O'Toole (2017), has a symbolic library to be found at <https://www.sympy.org/en/index.html> that could allow this programme also to be used as an automated theorem prover but the use of this library is very limited so far.

Live in-browser version of the [Lean theorem prover](#).

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert a,
4   intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 3:2**  
 $a b : \mathbb{N},$   
 $h : a = b$   
 $\vdash b = a$

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert a,
4   [intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 4:2**  
 $b : \mathbb{N}$   
 $\vdash \forall (a : \mathbb{N}), a = b \rightarrow b = a$

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert a,
4   intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 5:2**  
 $b a : \mathbb{N},$   
 $h : a = b$   
 $\vdash b = a$

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert a,
4   intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 6:2**  
 $b a : \mathbb{N},$   
 $h : a = b$   
 $\vdash a = b$

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert a,
4   intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 7:0**  
no goals

**Fig. 1** LEAN online interface - Multiple screenshots of a complete and correct proof illustrating the changes in the goal of the proof with each line of the written code

shows the goals or sub-goals required for completion of the proof and provides feedback to the user in terms of the symbol consistency and the logical connectedness of the statements. Figure 2 shows an error message as the user had inserted an incorrect parameter in the revert tactic. It has been proposed (Avigad, 2019) that engagement with LEAN could prove beneficial for students. Avigad describes an undergraduate course in elementary logic which introduces mathematical proof using LEAN.

Live in-browser version of the [Lean theorem prover](#).

```

1 example (a b : ℕ) (h : a = b) : b = a :=
2 begin
3   revert|z,
4   intros,
5   symmetry,
6   assumption,
7 end

```

**goal at 3:8**  
 $a b : \mathbb{N},$   
 $h : a = b$   
 $\vdash b = a$

**3:2: error:**  
get\_local tactic failed, unknown 'z' local state:  
 $a b : \mathbb{N},$   
 $h : a = b$   
 $\vdash b = a$

**Fig. 2** LEAN online interface - Error message appears due to incorrect use of the variable  $z$  in one of the LEAN tactics

The course did not require mathematical background higher than secondary school mathematics and focused on the three languages that are at play when the students engaged with the course: informal mathematical language, formal symbolic logic and the computational proof language. Avigad writes about the informal feedback from students who report being able to switch between the different languages and being able to overcome difficulties linked to the complexities of the LEAN syntax, but this author did not systematically investigate the proof written by the students or collect any systematic data.

Following the positive indications on the impact of LEAN on students' proof production suggested by Avigad (2019), this exploratory study investigates the characteristics of proofs written by students who engaged with the software. We ask the following research questions:

**RQ1:** What characteristics are observed to be common to proofs by students who engaged with the software LEAN?

**RQ2:** Are those characteristics common also in proofs by students who did not engage with the software LEAN?

## Theoretical Framework

In order to investigate students' proof attempts we adopt a theoretical framework introduced by Fukawa-Connelly (2012). This framework links proof writing to proof comprehension and focuses on aspects of both these activities. Below, we briefly report the components of this framework and we reference relevant literature on each of these components.

### Proof Writing

According to Selden and Selden (2007) a proof is divided in two parts. The first is a formal-rhetorical part where the student establishes the nature of the proof to be found, makes explicit the given parts of the statement which may be implicit and generally sets the goal of the proof in mathematical language (if it is not given explicitly that way). This stage serves to lay down all the instruments that will be needed for the proof to come. The second part of a proof, according to Selden and Selden (2007), is the problem-centred part. Here the proof starts in earnest, and the proof activity is treated like a problem-solving task. Following Selden and Selden, these proof parts contain different activities. For example, the formal-rhetorical part may contain statements of the definitions of the mathematical objects that are included in the proof statement. The problem-centred part is likely to contain the development of a coherent mathematical argument that has the statement to prove as an end goal, or an eventual subdivision of the proof goal in intermediate sub-goals.

### Proof Comprehension and Proof Writing

Fukawa-Connelly (2012) observes that the model of proof comprehension proposed by Mejia-Ramos et al. (2012) and Selden and Selden (2007) framework for proof

writing overlap significantly. By using these two frameworks, the author provides a list of students' actions related to successful proof comprehension and proof writing. These actions, following Fukawa-Connelly (2012), are described below, together with examples from the literature that has addressed these actions.

- **Definitions and their use.** This action consists in stating the definitions given in the text of the proof and those necessary to use but not included in the statement to prove. Fukawa-Connelly (2012) links this definition-stating action to the formal-rhetorical aspect of the proof in Selden and Selden (2007). The importance of correct use of definitions in proofs has been highlighted also by other researchers such as Moore (1994) who described the occasional lack of students' understanding of the definitions involved in a proof and the inability to use those definitions to achieve the main goal of the proof.
- **Mathematical symbols and their use.** This is the ability to manipulate mathematical symbols according to the rules of logic (e.g., correct use of quantifiers). Linked to these actions are the findings in Selden (2011) who describes students' difficulties with quantifiers, especially at the transition from school to university mathematics. More recently, Lew and Mejía-Ramos (2019) investigated the discrepancy between the expectations of mathematicians and mathematics students regarding the writing conventions of proof. They found three categories of discrepancies: that mathematicians thought proofs should follow the academic style of writing and that the writing should be correct and follow also the rules of grammar, that attention should be paid to the definition of new mathematical objects in proofs and that the level of formality required of a proof depends on the context in which the proof is situated.
- **Logical status of statements and their links.** This action consists in obtaining the correct relationship between statements on the chain of reasoning of the proof and being clear on what the proof has to accomplish (as in Selden and Selden (2007)). Amongst studies that have investigated the way in which students link statements in proofs there are those which highlight the difficulties students have to keep the technical (mathematical) and everyday reasoning and language separate (e.g., Cornu (1991) and Lee and Smith (2009)).
- **High level ideas.** Linked to the problem-centred aspect of Selden and Selden (2007) proof framework, it includes evidence that the student has grasped the main ideas behind the construction of a proof and the way to approach such proof construction tasks. This is akin to the need for key ideas described in Raman (2003).
- **Modular Structure of the Proof.** The ability to identify how the parts of the proof fit together, and where there are sub-statements that need to be proved separately, then bring them together to obtain the statement that was to be proved. The importance of being able to recognise the structure of a proof was also highlighted by Leron (1983) and it is at the core of the proof method he proposes.
- **Use of examples.** Examples have many roles in writing proof: from convincing the student that a statement is true to helping them develop a general strategy for a proof. This is a research area that has recently gained renewed attention

especially in connection to the role of generic examples in writing proof (e.g., Aricha-Metzer and Zaslavsky (2019)).

Each one of these actions indicates a step towards proof writing and comprehension, and the data in this study has been analysed by operationalising this framework to indicate what proof actions and characteristics of proof writing we could detect in the work of students who had engaged with the theorem prover LEAN.

## Methods

### Context

This paper draws on a dataset collected during an exploratory study concerning the use of LEAN for an introduction to proof module<sup>2</sup> offered during the Autumn term in the first year of the students' undergraduate studies at a research-intensive university in the UK. The module had a standard content including number systems, sets, permutations and combinations; the binomial theorem; equivalence relations and arithmetic modulo  $n$ ; Euclid's algorithm; and an introduction to limits. The content was taught in 29 one-hour lectures taking place three times a week. Included in the module were also weekly seminar classes with exercise sheets. These exercise sheets were written by the Professor who taught the module, a research mathematician interested in number theory and formal proof verification, so that they could be solved both traditionally with pen and paper and by using LEAN.

All students were offered the opportunity to attend voluntary workshops on programming LEAN organised by the Professor who taught the module. These workshops were held every week, during the teaching period, in the evening and had no time limit, lasting on occasions till late. The students who attended these workshops had access to online resources as well as the Professor's blog, and they could post on a dedicated online forum. During the workshops the students would engage in the exercises of their first-year module and formalise statements from other modules (e.g., Galois theory) aiming to enrich the LEAN library. They worked sometimes on their own and then shared their work, posted queries in the online forum, or worked collaboratively on the formalisation or proof of a statement. Some information about the nature and use of the software was also given during the lectures. The Professor would occasionally show how to use LEAN to prove some simple statements and explain how it could be useful in terms of giving instant feedback regarding the logical coherence of proofs or statements. Compared to the instruction situation described in Avigad (2019) and to the use that Dubinsky (1995) and others made of ISETL, the situation described in this paper was not a structured teaching intervention but rather an opportunity for further engagement with mathematics and programming offered to the students.

---

<sup>2</sup>University degrees in the UK are typically modularised. To progress from one year to the next, each student needs to accrue a certain number of credits by taking appropriate modules.

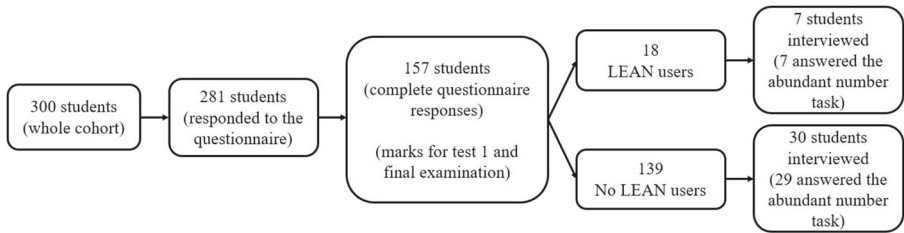


Fig. 3 Data collection breakdown

## Data Collection

Prior to the start of the study the authors applied for and obtained Ethics approval. Three-hundred students were registered on the module, nearly half of which were international students (i.e., non-UK or EU students). Assessment for the module consisted of five tests occurring every two weeks and a final examination in January. For the purposes of this paper we focus on the data presented in Fig. 3 where a detailed breakdown of the participants of the study is presented, and the informal observations of the evening workshops which happened during the teaching period.

The questionnaire was administered halfway through the teaching period to ascertain the ways in which students had engaged with LEAN during the first part of the semester. In this questionnaire, students were also asked whether they would be willing to take part in an interview and 37 students agreed. For the text of the questionnaire see Appendix 1.

The aim of the interviews was to gauge characteristics of the proofs of students who attended the voluntary workshops regularly and those who did not. Each interview lasted about one hour, was carried out by the first author of this paper and was audio-recorded and partially transcribed. The written materials the students produced during the interviews were also collected for analysis. The interviews started with some clarifications of the answers the students provided in the questionnaire to gain further information regarding their use and experience of LEAN. The students were then asked to engage in a series of proof tasks (see Appendix 2) involving familiar and unfamiliar mathematics definitions following a think aloud technique (Gillham, 2005). The focus of the analysis contained in this paper, a proof task involving an unfamiliar definition (Fig. 4) is justified by the evidence in the proof writing literature that in the context of familiar proofs students may resort to known procedures and may try to remember the proof as they had seen it instead of engaging with a proof as a problem-solving task (Azrou & Khelladi, 2019).

An *abundant number* is an integer  $n$  whose divisors add up to more than  $2n$ .  
A *perfect number* is an integer  $n$  whose divisors add up to exactly  $2n$ .

**Task:** If  $n$  is perfect, then  $kn$  is abundant for any  $k \in \mathbb{N}$ .

Fig. 4 The abundant number task



The main data analysed for this paper are the 36 students' written answers (7 LEAN users and 29 No LEAN users) to the abundant numbers task (Fig. 4) together with their corresponding interview extracts. One of the 37 interviewed students was not able to engage with this task due to time constraints.

At the start of the task the students were told that they had to decide on the truth or otherwise of the statement in Fig. 4 and, if they thought the statement was true, to provide a proof for it. If a student realised that the statement is false for  $k = 1$ , they were then prompted to provide a proof for  $k > 1$ .

## Data Analysis

This exploratory study aimed at finding characteristics of proofs by students who had engaged with the automated theorem prover and ascertain whether those were also common characteristics found in proofs written by students who had not engaged with LEAN. In order to do so we proceeded in three stages:

- **Stage 1:** Investigation of the cohort as a whole in terms of achievement at the start and at the end of the teaching period. We could not otherwise discard or confirm the hypothesis that the students who engaged with LEAN were higher achievers than their peers at the start of the module.
- **Stage 2:** Scoring the proofs of the abundant number task to create a finer categorisation based on 'achievement level' rather than on correct and incorrect outcome. Given that students who had engaged with LEAN did receive more informal mathematics instruction and mathematics time, it would not be reasonable to make claims related to achievement in the study.
- **Stage 3:** Qualitative coding the proofs which are scored at the same achievement level by analysing the written output together with the interview extracts.

Below, we describe how these three steps of data analysis were carried out.

### Stage 1: Students' Attainment Data

Students were categorised as No LEAN users (NL) or LEAN users (L) according to the response to one of the questions in the questionnaire asking whether they engaged with the voluntary workshops on LEAN. LEAN users were students who had consistently engaged in the voluntary evening workshops on LEAN, and No LEAN users were students who had either never engaged with LEAN or had taken part in one or two of the voluntary workshops and then stopped. Of the 281 students who completed the questionnaire, 157 gave information regarding their familiarity with LEAN. Among those 157 students, there were 18 LEAN users and 139 No LEAN users (Fig. 3).

The marks for the first test and the final examination for the 157 students who shared their familiarity with LEAN were analysed in order to capture differences in attainment between groups of students at the start (test 1) and at the end (final examination) of the teaching period. An ANOVA was performed between the attainments of the two groups on the examination results to calculate the contribution to the

variance of the uptake of the LEAN classes. A *t*-test was also performed on the means of the results of test 1 of the two groups to see whether the LEAN group performed significantly better than the No LEAN group already at this early stage.

## Step 2: Proof Scores

The second stage of the analysis consisted in scoring the 36 proofs of the statement in Fig. 4. In order to do so we employed a scoring scheme adapted from Zazkis et al. (2015). A proof scored 4 if complete, well organised and valid; 3 if valid, but with minor inaccuracies both regarding the language and the structure; 2 if not valid/complete but good progress had been made; 1 if some steps were attempted but no progress was made and 0 if the task was not attempted at all. This scoring gave an indication not only of the correctness of a proof, but also of the progress that was made towards a complete and correct proof. The proofs were scored independently by the two authors of this paper and a pure mathematician who agreed to help and was not aware of any of the background of the project, but only that these were proofs written by first-year mathematics students. After the independent scoring of the proofs, the ambivalent cases were discussed and a final score list was obtained.

## Step 3: Qualitative Analysis of the Interview Data

The unit of analysis of the interview data was the solution to the abundant number task (see also Table 2), which comprised both the written output and the verbal explanations given by the student during the interview. This is called the *interview output* throughout the paper. The analysis of the 36 interview outputs was divided into two phases. In the first phase each interview output was divided in a formal-rhetorical part and a problem-centred part, according to Selden and Selden (2007) framework. Within these two sections, characteristics were highlighted which are also mentioned in Selden and Selden (2007), such as making elements of the proof overt which are hidden in the statement of the theorem, often found in the formal-rhetorical part. In the second phase of the analysis the interview outputs were coded line by line using codes originating from an operationalised version of the categories in Fukawa-Connelly (2012) as described in Table 1. Throughout the coding exercise no further codes were used, as there were no necessary additions to the ones appearing in Table 1.

**Table 1** Operationalised version of the categories in Fukawa-Connelly (2012) with code labels used in the analysis of the interview outputs

Proof outputs coding scheme	
Definition Given [DEF-GIV]	Links to previous statement [LOG-FOL]
Using familiar definitions [DEF-FAM]	High-Level ideas [APP]
Using familiar statements/terms [STAT-FAM]	Modular structure of the proof [FIT]
Introduction of symbolism [SYM]	Example Use [EX]
Logical status of statements [LOG-STAT]	

**Table 2** Example of coding of a proof of the abundant number statement produced by the authors

Mathematical step	Coding	Comments
[1] False for $k = 1$		Limit case
[2] Let $n$ be a positive perfect integer	[DEF-GIV]	Formal -Rhetorical phase
[3] Let $d_1, d_2, \dots, d_r$ be the divisors of $n$	[DEF-GIV] [STAT-FAM]	
[4] We have $\sum_{i=1}^r d_i = 2n$	[DEF-GIV]	
[5] Consider now $kd_1, kd_2, \dots, kd_r$ these are also among the divisors of $kn$	[SYM], [LOG-FOL]	Problem solving part
[6] We have that $kd_i \neq 1$ for each $i$ as $k > 1$	[LOG-STAT] [LOG-FOL]	The prover is creating a path
[7] 1 will also be a divisor of $kn$	[LOG-FOL]	
[8] As $\sum_{i=1}^r kd_i = 2kn$	[LOG-FOL]	(reference to strategic knowledge)
[9] $\sum \text{divisors of } kn \geq \sum_{i=1}^r kd_i = 2kn + 1$ $\quad > \sum_{i=1}^r kd_i$ $\quad = 2kn$	Conclusions	
[10] This implies that $kn$ is abundant		(Intuition)

A small number of interview outputs were coded independently by the two authors and then reviewed so that an agreement on code meaning was found. The first author of the paper completed this coding exercise. An example of application of this framework is in Table 2 where a proof of the abundant number task, created by the authors, is coded.

Students’ responses were grouped according to their score and then examined qualitatively to find similarities and differences between No LEAN and LEAN users within the groups that received the same score.

## Findings

### Step 1: Students’ Background

When we asked whether the two groups performed significantly differently in the first test we found that the mean performance of the No Lean (NL) group was 7.1 and the Lean (L) group was 7.8. This is (just) not significant ( $t(155) = 1.68, p = 0.10$ ). However, the variances were not equal, so a Wilcoxon test was used to compare medians. This test returned non-significance:  $W = 1125, p = 0.49$ .

After adjusting for performance on test 1 administered in the second week of the teaching period, and which was significantly related to the final examination ( $F(1, 154) = 8.60, p = 0.004, r = 0.23$ ), there was a significant effect for being in the group that had taken part in the LEAN voluntary classes (L group) on the examination score ( $F(1, 154) = 9.60, p = 0.002, \text{partial } \eta^2 = 0.06$ ). That is quite a small effect and about 6% of the variance in the examination scores can be accounted for

**Table 3** Final scoring of the proofs of the abundant number task - number of students (frequency percentages)

Scores	No LEAN users	LEAN users
4	4 (13.8%)	3 (43%)
3	3 (10.3%)	3 (43%)
2	5 (17.3%)	0 (0%)
1	16 (55.2%)	1 (14%)
0	1 (3.4%)	0 (0%)

by being in the L group (which is around about the same as the amount of variance accounted for by the variance in scores on test 1).

From this analysis, we can infer that the group who had attended the LEAN voluntary classes did indeed perform better in the final examination than the group of students that did not attend those classes regularly. This result was to be expected as those students engaged in more mathematics workshops than their counterparts, and did so voluntarily, therefore they were likely to engage meaningfully with those workshops. What is relevant to our analysis is that those students did not perform significantly better to start with - as shown by the analysis of the results of the first test early on in the teaching period. Therefore we can reasonably exclude the hypothesis that the students who took part in the LEAN workshops were higher achievers compared to their peers when they joined the university.

## Stage 2: Proof Scores

Table 3 reports the final scoring of the 36 proofs on the abundant number task. The results indicate that the students who engaged with LEAN indeed wrote proofs that scored higher on average than the proofs written by the other students. Reading this scoring, along with the results of the statistical analysis presented earlier, suggests that a better performance on proof was to be expected, as the interviews during which the data was collected were carried out towards the end of the teaching period.

## Stage 3: Qualitative Analysis of the Proof Outputs

In this section, we report the qualitative analysis of the interview outputs. We do so by focusing on three groups: students who scored 4, 3 and 1 in the scoring exercise. We will not include students who scored 2 and 0 as none of the LEAN users obtained such scores.

### Successful Proofs: Score 4

Seven students scored 4 on the abundant task proof: three LEAN and four No LEAN users.<sup>3</sup>

<sup>3</sup>The student names are pseudonyms, and on occasion the gender of the participant has been changed. Names starting with L denote LEAN users and with N denote No LEAN users. For reasons of space we cannot reproduce the proof with the relevant transcript and coding for all examples included in the paper.

$$n \text{ is perfect} \Rightarrow \sum_{j=1}^i d_j = 2n \quad \left| \text{let } n \text{ have divisors } d_1, d_2, \dots, d_i \text{ for some } i \in \mathbb{Z} \right.$$

Fig. 5 LEAN user student Leonardo - formal-rhetorical part of the proof

Proof: let  $n$  be a perfect number.  
~~then~~  
then, the sum of the divisors of  $n$ , call them say  $x_1, x_2, \dots, x_i$   
are equal to  $n$   
 $x_1 + x_2 + \dots + x_i = n$   
with  $x_i \neq x_j$   
 $\forall i, j \in \mathbb{N}$

Fig. 6 No LEAN user student Nathan - formal-rhetorical part of the proof

Figures 5 and 6 show the formal-rhetorical part of the proofs produced by Leonardo and Nathan who scored 4 in the scoring exercise. Both students start their proofs by unpacking the definitions given in the statement (perfect number), which is a characteristic of the formal-rehetorical part of the proof. The coding of these sections consists only in codes related to definitions (e.g., [DEF-...], see Table 1). The students assign symbols to the main objects in the definition of perfect number (e.g., the divisors) and express algebraically the relation between the perfect number and its divisors. However, the accuracy of mathematical writing varies between the two proofs. While Leonardo accurately uses symbolic language and logical connections to define a perfect number, Nathan does so to a lesser extent. In Leonardo’s writing,

If integer  $n$  is a perfect number  
we assume  $n$  has divisors  
 $(r_1 \leq r_{i+1} \quad \forall i \in (1, m-1))$  s.t.  $r_1, r_2, \dots, r_m$  ( $r_i \neq r_j$  if  $i \neq j$ )  
 $r_1 + r_2 + \dots + r_m = 2n$

Fig. 7 No LEAN user student Ned - formal-rhetorical part of the proof

Lemma ①  $n | kn \quad k, n \in \mathbb{Z}$   
 Lemma ②  $n | p \wedge d | n \Rightarrow d | p, \quad d, n, p \in \mathbb{Z}$   
 By lemma ①  ~~$d_i | n \Rightarrow d_i$~~   
 Lemma ③  $d | n \Rightarrow d | kn$   
 Assume  $d | n$ ,  ~~$p = kn$~~  apply lemma ② with  $p = kn$   
 Apply lemma ①  $\Rightarrow d | kn$ .  
 By lemma ③  ~~$\forall i \in \mathbb{Z}$~~  For all  $1 \leq i \leq n, i \in \mathbb{Z}, d_j | kn$ .  
 Lemma ④  $d | n \Rightarrow kd | kn$ .  
 By lemma ④ For all  $1 \leq j \leq i, j \in \mathbb{Z}, kd_j | kn$ .  
 $\Rightarrow \sum_{j=1}^i \text{divisors of } 2kn \geq \sum_{j=1}^i kd_j = k \sum_{j=1}^i d_j = 2kn$ .  
 Cases with  $k=1$  and  $k \neq 1$ .  
 $2kn < 2kn + 1 \Rightarrow \dots \leq \sum \text{divisors}$

Fig. 8 LEAN user student Leonardo - formal-rhetorical part of the proof

mathematical symbols are used either to signal the logical connections between the property of  $n$  being perfect and the relationship between the number and the sum of divisors which is  $2n$  (the use of the logical implication symbol) or to illustrate the sum of the divisors. Inaccuracy on the use of mathematical language and mathematical writing is common to other proofs that scored 4 and were written by students who did not engage with LEAN. Figure 7 is one such example.

In Fig. 7, we can see that Ned lays out all the elements for the central part of the proof, but the writing is somewhat confused and the use of mathematical symbols is, at times, ambiguous.

Figures 8 and 9 show the problem-centred parts of the proofs written by Leonardo and Nathan. Characteristic of Leonardo's writing is the subdivision of the proof in lemmas which will each need to be proved before the proof is complete. Each of these steps was coded [FIT] to highlight the modular structure of this proof. Indeed, these lemmas represent the goals and sub-goals of this proof, providing Leonardo's proof with a clear structure. The sequence of sub-goals results in illustrating that  $kn$  is an abundant number by taking cases of  $k = 1$  and  $k \neq 1$ . As for the previous part the student makes much use of mathematics symbols and is also concerned, once a class of objects are introduced, with defining exactly where the objects belong to. This is clear in the start of the proofs of Lemma 3 and Lemma 4. Figure 9 shows the final outcome of the final proof written by Nathan. In this case the student had resorted to examples first of all to understand how the proof could be written, e.g., to gauge what were the relevant characteristics of mathematical objects involved.

let's consider  $kn$ , where  $k \in \mathbb{N}$  and  $k > 1$ :  
 and and call  $kn = y$   
 $y = kn = k(x_1 + x_2 + \dots + x_i)$   
 $= kx_1 + kx_2 + \dots + kx_i$   
~~We know that  $y$  is equal to  $kn$~~   
 We know that  $kx_i \neq kx_j \quad \forall i, j \in \mathbb{N}$   
  
 We know  $y$  is equal to  ~~$kx_1 + kx_2 + \dots + kx_i = y$~~   
 ~~$kx_1 + kx_2 + \dots + kx_i = y$~~   
  
~~But this is~~  
 but we know that 1 is not one of  
 $kx_i \quad \forall i \in \mathbb{Z}$   
 because  $k > 1$  and  $x_i \in \mathbb{Z}$   
  
~~Therefore~~ 1 is certainly a factor of  $y$ .  
 and hence the sum of the factors of  $y$  is  
 at least  $y + 1$ .  
 $y > k \cdot y$   
 by definition of abundant number  
 $y$  is abundant.  $\square$

Fig. 9 No LEAN user student Nathan - formal-rhetorical part of the proof. (During the interview the students were asked to first produce a proof and then using a different coloured pen to add something further if they wished - Nathan is one of the students who did use this option)

Once he convinced himself of the truth of the statement and the structure of the proof via examining some examples, he produced the proof in Fig. 9. The first part of the transcript, together with the corresponding writing, was coded with [EX] codes mainly. For the final write up of the proof Nathan (Fig. 9) starts by limiting the values that  $k$  can take excluding the case  $k = 1$  but not saying explicitly that the statement is not valid in this case. He then replaces  $kn$  and  $n$  with the sum of their divisors (since  $n$  is perfect) and moves on to multiply each one by  $k$ . He continues by saying that 1 is

Note that  $k a_1, k a_2, \dots, k a_n$  are therefore all divisors of  $kn$ .  
 And we have  $k a_1 + k a_2 + \dots + k a_n = 2kn$ .  
 Factors of  $2kn$  include 1, which  $\{k a_1, k a_2, \dots, k a_n\}$  does  
 not include ~~because~~ because  $a_i \geq 1$ , and  $k > 1$ , so  $k a_i > 1$ .  
 So the set does not contain 1.  
 But  $1 + k a_1 + k a_2 + \dots + k a_n = 2kn + 1 > 2kn$   
 By definition  $kn$  is abundant.

Fig. 10 LEAN user student Louis - problem centred part of the proof

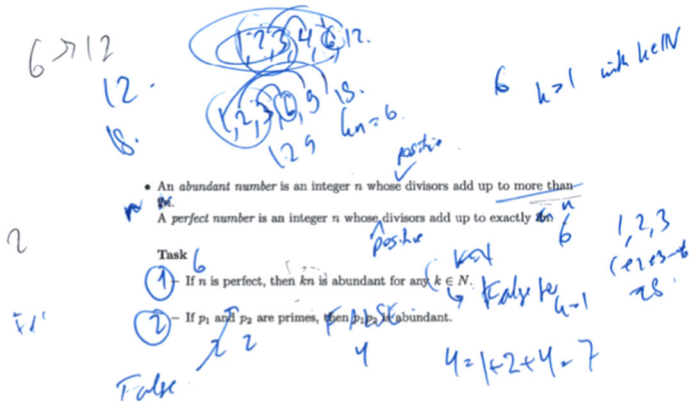


Fig. 11 No LEAN user student Nathan- example use

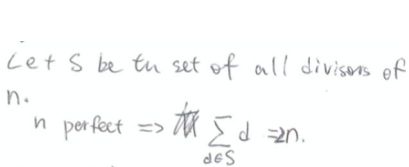
also a divisor of  $kn$  so  $kn$  is abundant. This is a correct proof, but there is no obvious signposting of sub goals (codes [FIT]) that need to be proved and the mathematical writing is not as accurate as the case of Leonardo’s proof.

Louis, another student who attended regularly the LEAN workshops, shows similar characteristics in his proof to those we can see in Leonardo’s writing, as we see in Fig. 10. The proof takes more of a narrative form here, characterised also by the presence of punctuation, the language is precise and the definitions are used appropriately in the proof.

We conclude this section by noting the example use of the students who scored 4 in the proof scoring exercise. We do so here, and in the rest of the qualitative analysis, as example use is the last category of proof comprehension and proof writing that is included in our operationalised version of the Fukawa-Connelly (2012) framework (see Table 1).

Of the seven students who scored 4, five used some example when thinking about this proof: of those five, two were LEAN user and three were not. Noticeably Nathan dedicated significant time to work out examples for this proof - as we can see in Fig. 11.

The examples used by these students were all at the start of that section of the interview and were used to understand the statement of the task by trying a decomposition in factors of one or two perfect numbers and a few values of  $k$ . The numbers that were



[10] For a  $k$  larger than 1, I think it would be true. Shall I prove that one then?

[11] Let ... I should just use a set. Let  $S$  be the set of divisors of  $n$  [SYM]

[12]  $n$  perfect implies that ... Oh add up, [DEF-GIV]

[13]  $d$  is a member of  $S$ ,  $d$  equals  $n$  ... alright so ... [DEF-GIV] and [LOG FOL]

Fig. 12 LEAN user student Laura’s Formal-Rhetorical part



- if  $d|m \Rightarrow kd|nk$   
 we know since  $n$  is perfect  $\sum_{d|m} d = 2m$   
 so  $\sum_{d|kn} d \Rightarrow \sum_{d|m} kd = 2kn$

- [2] I am thinking what I am trying to do is to find a good way to describe the divisors of  $n$ . [SYM]
- [4] Yeah, I will just go with a sequence, like he has used a sequence before. [STAT-FAM]
- [5] So let ... wait a second for any  $k$  ... hm ... What if  $k = 1$ . [LOG FOL]

Fig. 13 LEAN user student Lydia’s Formal-Rhetorical part

chosen by these students were 6 (the smallest perfect number), and then 28. These numbers, which were often accompanied by checking that the sum of their divisors was double the original numbers, seem to help the students to start constructing a (successful in this case) proof.

**Valid Proof with Minor Errors - Score 3**

Six students scored 3 on the abundant number task proof, three LEAN users and three No LEAN users. Figures 12 and 13 are the formal-rhetorical parts of Laura’s and Lydia’s proofs - both LEAN users.

This presence of text (e.g., ‘Let  $S$  be the set of all divisors of...’) in Laura’s proof is typical of the whole proof, as well as of proofs of other LEAN users as we can see in Lydia’s proof (‘we know since  $n$  is perfect ...’). In this stage there is also attention to setting symbolism (codes [SYM]) in a helpful way so that it can be successfully used to proceed through the second phase of the proof, naming the main mathematical objects that are involved in the proof.

Nataly’s writing in the formal-rhetorical part (Fig. 14) shows that initially the idea of prime factorisation was considered and then abandoned as not so helpful (see transcript - Fig. 14). Then, the sum of divisors is introduced without a clear statement

$n = \prod_{i=1}^n p_i^{k_i}$   
 $\Rightarrow S = \sum p_i$   
 $S_n = 2n = \sum p_i \quad \forall p_i | n$

- [1] Well we know that  $n$  can be written as the product from  $i$  is 1 to  $n$  of  $p_i$  to the  $k_i$  because this is the fundamental bit of arithmetic which implies that sum of factors which we are going to call  $S$  is ... [DEF]
- [2] Ah ... but we can’t use that because we need to take all the combinations. So perhaps this method is not the right way of doing it. [DEF and LOG-FOL]
- [3] I think I am struggling with notation a bit.
- [4] Okay so we will start off with  $S$  little  $n$  is equal to  $2n$ . Ahm ... So  $S_{kn}$  is  $k$  times each factor ... I can understand the essential argument but writing it down is ... a bit of a problem

Fig. 14 No LEAN user student Nataly - formal-rhetorical part of the proof

$$\begin{array}{l}
 1. \quad n: d_0, d_1, d_2, \dots, d_r \quad \leftarrow \begin{array}{l} 1, 2, 4, 7, 14, 28 \\ 1 + 14 \end{array} \\
 2n = d_0 + d_1 + d_2 + \dots + d_r \quad \leftarrow \begin{array}{l} \text{where } d_i \text{ are} \\ \text{divisors of } n, 0 \leq i \leq r \end{array} \\
 kn: d_0, d_1, d_2, \dots, d_r, e_0, e_1, \dots, e_s \\
 \text{sum } \alpha
 \end{array}$$

Fig. 15 No LEAN user student Norman - formal-rhetorical part of the proof

as to what each of the involved symbols mean. Indeed, in the next line, Nataly states that what she wrote is valid for each  $n$  but does not clarify the set where  $n$  belongs to. This may be not important at these early stages of the students' mathematical instruction, but it may become important later on when proofs become more complex. Indeed, in this extract there are no codes referring to setting symbolism ([SYM]) and difficulties with setting efficient notation is also acknowledged in the interview (Fig. 14).

This lack of precision in introducing mathematical symbolism is common to other students, as we can see from the extract from the proof by Norman (Fig. 15), another student who did not engage with the LEAN workshops.

While it is possible to classify these sections of the proofs (Figs. 13, 14 and 15) as formal-rhetorical in the sense of Selden and Selden (2007) as they contain unpacking the definitions that are present in the statement, it is also easy to see that the ones written by Laura and Lydia are more organised and use mathematical language more precisely than the one written by Nataly or indeed Norman.

We consider now the problem-centred parts of the proof for those students who scored 3. Laura's problem-centred part shows again attention to correct use of symbols (Fig. 16). One such example is the change of letter to indicate the divisors of  $kn$  so that there will be no confusion with the divisors of  $n$  which were named earlier. Also, interesting are the reliance on symbolic mathematical language, clarity in the structure of the proof and most noticeably the written explanations given at each step. Present is also a division in two sections which illustrate two sub-goals of the proof: the first is showing that 1 is not included in the set  $kS$  and the second showing that the sum of divisors of  $kn$  is greater than  $2kn$  ([FIT] code). This may indicate clarity in understanding of how the proof should be organised but also care for the clarity of this proof.

Nataly's proof on the other hand (Fig. 17) consists of a chain of equalities and inequalities without written explanation of any sort. The chain of inequalities coded only with codes [LOG-FOL] leads eventually to a justification of the statement that she intended to prove but does not show clarity of writing or attention to correct symbolism. This lack of attention to structure and writing is visible in proofs written by other No LEAN users, as also the extract from Norman shows, see Fig. 18.

Let  $S$  be the set of all divisors of  $n$ .  
 $n$  perfect  $\Rightarrow \sum_{d \in S} d = 2n$ .  
 Let  $kS$  be the set  $\{k \cdot d \mid d \in S\}$   

$$\sum_{d \in kS} d = \sum_{d \in S} kd$$

$$= k \sum_{d \in S} d$$

$$= 2kn$$
  
~~Since~~  
 Let  $m$  be an arbitrary element of  $kS$ .  
 ~~$k \cdot d$~~   $d$  s.t.  $m = d \cdot k$  where  $d \mid n$   
 $k \mid m$   
 $\Rightarrow d \cdot k \mid m \cdot k$   
 $\Rightarrow m \mid kn$ .  
 $\therefore \forall m \in kS (m \mid kn)$ .  
 Since  $\exists k \neq 1, 1 \notin kS$ .  
 Let  $T$  be the set of all divisors of  $kn$ .  
 Since  $\forall m \in kS, m \mid kn$ .  
 $kS \subseteq T$ .  
 Since  ~~$1 \notin kS$~~   $1 \mid kn$  but  $1 \notin kS$   
 $kS \subset T$ .  

$$\sum_{d \in T} d \geq 1 + \sum_{d \in kS} d$$

$$> \sum_{d \in kS} d$$

$$\Rightarrow 2kn$$
  
 By definition of abundant number  $kn$  is abundant.

- [11] Let ... I should just use a set. Let  $S$  be the set of divisors of  $n$  [SYM]
- [12]  $n$  perfect implies that ... Oh add up, [DEF-GIV]
- [13]  $d$  is a member of  $S$ ,  $d$  equals  $n$  ... alright so ... [DEF-GIV] [LOG-FOL]
- [14] Let  $kS$  be ... how should I describe it? [SYM]
- [15] The set so each element of  $kS$  would come from  $S$  but they are multiplied by  $k$ . [LOG-FOL]
- [16] So  $k$  times  $d$  where  $d$  is a member of  $S$  [LOG-FOL] I think that is a good enough notation
- [17] So  $d$  coming from  $kS$  that should equal  $kn$  because its multiplied ...
- [18] Yeah because ... that was probably unnecessary. So now I have to prove that every element of  $kS$  is a divisor of  $kn$ . [FIT]
- [19] Okay. How do I prove that? ... which letter I haven't used? Letter  $m$ ? [SYM]
- [20]  $m$  arbitrary element of  $kS$  there exists  $k1$  such that..no it is not  $kn$  (deletes) [LOG-FOL]
- [21]  $d$  such that ...  $d$  times  $k$  so  $k$  ... sorry  $d$  divides the original  $n$  so  $d$  divides  $n$  implies that  $d$  times  $k$  divides  $n$  times  $k$  [LOG-FOL]
- [22] Implies that  $m$  divides  $kn$  that is for every  $m$  in the set  $kS$ .  $m$  divides  $kn$  okay [LOG-FOL]
- [23] So now we have proven that  $kS$  is a subset of the divisors  $kn$ ,
- [24] so since 1, since  $k$  does not divide 1 since  $k$  is larger than 1 ah ... 1 is not in the set  $kS$ . [LOG-FOL]
- [25] Okay. Can I use another paper? Let  $T$  be the set of divisors of  $kn$  [FIT]
- [26] Since for every  $m$  in  $kS$ ,  $m$  divides  $kn$  so  $kS$  is a subset of  $T$  [LOG-FOL]
- [27] since 1 apparently is a divisor of  $kn$  but 1 is not in the set  $kS$ . [LOG-FOL]
- [28]  $kS$  is a proper subset of  $T$  [LOG-FOL]
- [29] So the sum of, the sum of divisors of  $kn$  is larger than or equal to 1 plus the sum of divisors which is ... [LOG-FOL]
- [30] Yeah okay I think that this is good enough proof
- [31] By definition of abundant numbers  $kn$  is abundant [DEF-GIV]

Fig. 16 LEAN user student Laura formal-rhetorical part

Of the six students who scored 3 in the scoring exercise, only two used examples during this part of the interview, one LEAN user and one No LEAN user. The remaining four did not use examples at all and started the proof straight away. Again, the examples (the number 8 and its decomposition for the LEAN user - although he found that 8 was not a perfect number he did not continue with further examples, the number 28 and its decomposition for the No LEAN user) were at the start of the proof.

$$\begin{aligned}
 S_n &= 2n = \sum_{i=1}^n f_i \quad \forall f_i | n \\
 kS_n &= kf_1 + kf_2 + kf_3 \dots + f_1 + f_2 + f_3 \dots \\
 &= kS_n + S_n \\
 &= (k+1)S_n \\
 &> 2S_n \quad k \in \mathbb{N} \Rightarrow k \geq 1
 \end{aligned}$$

- [5] Because this is  $k$  times each factor. Is the sum of  $k$  times  $f_1$  plus  $kf_2$  plus  $kf_3 \dots f_1$  plus  $f_2$  plus  $f_3 \dots$  where these are all factors. [LOG-FOL]
- [6] This is  $kS_n$  plus  $S_n$ . [LOG-FOL]
- [7] Which is  $k$  plus 1  $S_n$  which is greater than  $2S_n$  as  $k$  is in  $\mathbb{N}$  and this implies that  $k$  is greater or equal to 1. I think  $k$  is not 1. Because otherwise you are saying  $n$  is perfect then in  $n$  is perfect. So  $k$  is greater than 2. [LOG-FOL - DEF]
- [8] I think that is the first part done.

Fig. 17 No LEAN user student Nataly - problem-centred part of the proof

**Unsuccessful Proofs: Score 1**

Seventeen students scored 1 in this task, one LEAN user and sixteen No LEAN users. Figure 19, shows the formal-rhetorical part of Luke’s proof, a LEAN user, who rewrites the definitions and introduces symbolism useful for the next part of the proof. From the transcript it appears that the process of taking the definitions in words and rewriting them in symbolic mathematical language is considered a sort of translation by the student, denoting awareness of the shift needed between non-technical to technical mathematical language (and eventually to programming language when relevant). Moreover, this student, as the other LEAN users, is very careful with explaining where various mathematical objects belong (e.g.,  $n$  in  $\mathbb{Z}$ ).

For Noah (Fig. 20) the formal-rhetorical part consists only of the prime factorisation of the number  $n$  and the sum of its divisors, without any explanation to accompany it. Note, also, that Noah uses the same notation to mean both the prime factors and the divisors. Although Noah does not write (or say) that  $d_i$ ’s are prime

$$\begin{aligned}
 & \dots \dots \dots, 0 \leq i \leq r \\
 & kn: d_0, d_1, d_2, \dots, d_r, e_0, e_1, \dots, e_s \\
 & \text{sum of} \\
 & \text{consider all divisors of } kn: \underbrace{d_0 + d_1 + \dots + d_r}_{2n} + \underbrace{e_0 + e_1 + \dots + e_s}_{\text{where } e_j \text{ are divisors of } k, 0 \leq j \leq s.} \\
 & 2n + e_0 + e_1 + \dots + e_s
 \end{aligned}$$

Fig. 18 No LEAN user student Norman - problem-centred part of the proof

Def - Abundant.  
 $\forall n \in \mathbb{Z}, n \text{ abundant} := \sum_{d|n} d > 2n.$   
 Def - Perfect.  
 $\forall n \in \mathbb{Z}, n \text{ perfect} := \sum_{d|n} d = 2n.$

$kn \in \mathbb{Z}$   
 $\sum_{d|kn} d = \sum_{d|n} d + 2n$

[6] Translating the definitions. Let's go. [SYM - DEF]  
 [7] For all  $n$  in  $\mathbb{Z}$ ,  $n$  abundant is defined as ... yeah adds to more than  $2n$  [DEF] [8] Cool. So if  $n$  is perfect then  $kn$  is abundant for any ... (restates the DEF) Right hm. Once side for thoughts and one side for proof.  
 [9] So  $kn$  equals by definition yeah ... is abundant ... (Continues to restate the DEF)  
 [10] Right divisors of this. Divisors of  $k$  times  $n$ . Hm? What what am I writing that is not true. I am trying to figure out divisors of  $k$  times  $n$ . [LOG FOL] Divisors of  $k$  times  $n$ . Oh my gosh this is horrible. [11]  $d$  can divide  $k$ ,  $d$  can divide  $n$  yeah but that doesn't help. [LOG FOL]

Fig. 19 LEAN user Luke's formal-rhetorical part

factors, this is necessary for the first equality in Fig. 20 to hold. This initial part of the proof is still coded as formal-rhetorical as there is an attempt to unpack the definitions included in the statement, but this unpacking is confused by a haphazard use of symbols.

As for the problem-centred parts, Luke's proof is unfinished (Fig. 21). Luke starts defining and discussing the notation and the different combinations of the divisors (the ones that divide  $kn$  but not  $n$  and other combinations) but he struggles to formalise the first part of his summation and is therefore unable to continue. This section of the transcript is coded as [LOG-FOL] to convey that fact that Luke tries to deduce via logical inferences the next steps of the proof but fails. Indeed, Luke's attempt to formalisation is extreme for this simple proof and his introduction of some computer science-type notation like the Iverson bracket (which is visible in his writing) does not help his attempt. Moreover, there are no codes related to modularisation of the proof as if the student has perhaps lost sight of what the proof was requiring them to achieve.

Noah's work (Fig. 22) continues with the prime factorisation, which is then abandoned in favour of trying to find a connection between the sum of factors and  $k$ . The codes of this part consist again in [LOG-FOL] to signify the presence of a chain of

$n = \prod_{i \in \mathbb{N}} d_i$   
 $kn = \prod_{i \in \mathbb{N}} d_i \times k$

$\sum_{i \in \mathbb{N}} d_i = 2n$   
 $2n + k$   
 $\sum_{j \in \mathbb{N}} d_j \rightarrow 2n + k = 2kn$

[1] I don't have any leads on the first part but I will try the first part to start with  
 [2] Let's try  $n$  equals the product of  $d$  - for divisor -  $d_i$ ,  $i$  equals. [DEF]  
 [3] Let's say  $i$  integer I mean natural numbers. Then  $kn$  is equal to the product of the divisors times  $k$ . [LOG FOL]  
 [4] And because by assuming that  $n$  is perfect here. We say the sum of factors/divisors are equal to  $2n$  [LOG FOL]

Fig. 20 No LEAN user student Noah's formal-rhetorical part

$kn \in \mathbb{N}$   
 $\sum_{d|kn} d = \sum_{d|n} d + \sum_{d|kn} d$   
 $= 2n + \sum_{d|kn} d$   
 $\geq 2n$

$[P] := \begin{cases} 1, & P \text{ true} \\ 0, & P \text{ false} \end{cases}$

$\sum_{d|kn} d = \sum_{d|n} [d|kn] d$   
 $= \sum_{d|n} [d|kn \wedge (d|n \vee d|kn)] d$   
 $= \sum_{d|n} [d|kn \wedge (d|n \vee d|kn)] d$

$d|n \wedge d|kn$        $d|n$   
 $(A \wedge B) \rightarrow (A \rightarrow B)$

$[A \wedge B] = B$

$d|kn \wedge (d|n \vee d|kn)$   
 $= (d|n \wedge d|kn) \vee (d|n \wedge d|kn)$

- [9] So  $kn$  equals by definition yeah ... is abundant
- [10] Right divisors of this. Divisors of  $k$  times  $n$ . Hm ... What am I writing that is not true. I am trying to figure out divisors of  $k$  times  $n$ . [LOG-FOL]
- [11] Divisors of  $k$  times  $n$ . Oh my gosh this is horrible.
- [12]  $d$  can divide  $k$ ,  $d$  can divide  $n$  yeah but that doesn't help. [LOG-FOL] I want to link it to the divisors of  $n$ . [LOG-FOL] Well if there are divisors of  $n$  then they will be divisors of  $kn$  [LOG-FOL].
- [13] Yeah so this will be equal to some stuff plus divisors of  $n$  [LOG-FOL]. But  $n$  is perfect [DEF]. So this is equal to some stuff plus  $2n$ . [DEF APP]
- [14] But this is clearly larger than ... or equals ... is a strict inequality for any  $k$ . [LOG-FOL]
- [15] Is zero in the naturals for this question? (...)
- [24] I just need to formalise that mess. (...)
- [33] I reckon they would accept that. But now I am not satisfied myself because I haven't formalised what that mess is and I know LEAN wouldn't take that. Yeah

Fig. 21 LEAN user student Luke's formal-rhetorical part

deductions which - in this case - are incorrect and do not lead to a proof. Noah then resorts to examples (codes [EX]) to try to find the pattern between the various values for  $k$  and the sum of divisors comparing  $2kn$  and  $2n + k$ . However, this attempt is also unsuccessful. One of the features of this and other unsuccessful proofs of No LEAN users is the presence of codes related to stating definitions in the problem-centred part, till the end of the attempt, and the absence of [FIT] codes that signify the presence of a modular structure to the proof. Moreover, in this as in other unsuccessful transcripts, after the problem-centred phase comes to an end we notice codes related to example use; as if the student resorted to examples to clarify the structure of the proofs when they realise that their previous proof attempt were flawed.

Finally, of the seventeen students who scored 1 only six No LEAN users used some examples. Of interest here is the fact that of these six, four resorted to examples after a failed proof attempt but were unable to continue and two started with some numerical examples but were unable to continue by producing a proof. Also of interest here is that Luke, the only Lean user who scored 1, does not resort at any point to examples in his proof attempt, remaining occupied mostly by issues related to notation.

~~$kn \Rightarrow$~~  if  $kn$  perfect  
 $2kn$   
 $2kn \geq 2n$   
~~but we know~~  
 Compare  
 $2kn$        $2n+k$   
 $k=1$      $2n$        $2n+1$   
 $k=2$      $4n$        $2n+2$

[16] I feel that I may have messed up and now I am stuck. I said that the sum of the divisors of  $kn$  is greater than  $2n + 1$  where I added  $k$  to  $2n$  where  $n$  was perfect. [LOG FOL]

[17] But now I am thinking if  $k$  was abundant the sum of its factors would be  $2kn$ . [DEF, LOG FOL]

[19] I will compare  $2kn$  to  $2n + k$  as ... as they are the sum of the factors, where  $2kn$  is the sum of the factors for a perfect  $kn$ . [LOG FOL] [DEF]

[20] Now  $2n + k$  is the sum of the factors for  $kn$  for, whether it is per-, abundant or not. [LOG FOL]

[21] So I will try numbers, values of  $k$ .  $k = 1$   $2n$ ,  $2n + 1$ . [EX]

[24] Would you mind if I just move on (Stops with the attempt of proving it)

Fig. 22 No LEAN user student Noah’s formal-rhetorical part

### Discussion

Recall that this exploratory study concerned the investigation of common characteristics of proofs written by students who had engaged in voluntary workshops on the use of the automated prover LEAN compared to proofs written by students who did not engage with such workshops. The motivation for the study was the conjecture found in the literature (Avigad, 2019) that engaging students with automated proof software would allow them to become more attuned to the requirements of rigour, which is characteristic to university mathematics. To this aim we compared qualitatively the proofs that had scored equally in a scoring exercise in order to find characteristics common to those written by students who engaged with LEAN and by those who did not.

We found two characteristics consistently observed in proofs produced by LEAN users which are only occasionally found on proofs produced by no LEAN users: one concerning the mathematical writing and one concerning the organisation and structure of the proofs. We discuss these in turn below and we add a note on the students’ example use.

**Mathematical Writing** LEAN users dedicated much effort to use technical mathematical language and symbolism correctly and they were very careful to state explicitly where certain mathematical objects belonged (e.g., divisors belonging to  $\mathbb{Z}^+$ , indices of a sum also belonging to  $\mathbb{Z}^+$  and being distinct - as we can see for example in the work of Leonardo, Fig. 8). These students showed awareness that this precision in mathematical language is important. Indeed Luke, another student who attended the LEAN workshops regularly, states in the interview that writing the proof is a sort of translation, making a distinction between technical language (used in mathematics) and non-technical language. Note that learning to distinguish between different language registers as an outcome of the use of LEAN was also hypothesised by

Avigad (2019). Moreover, this awareness of the distinction between technical and non-technical language is very important in the transition to university mathematics, as Cornu (1991) showed, and may originate from having to switch from mathematical language to coding language when using LEAN.

Also noticeable in proofs written by LEAN users is the precise introduction of the mathematical objects that play a role in the proof (in this case perfect numbers). This theme was also found in Lew and Mejía-Ramos (2019) as a requirement, by the mathematicians interviewed for that study, of a correct proof. This requirement is reflected in the data by the fact that the formal-rhetorical parts of proofs written by LEAN users contain only codes connected to definitions as these are used for setting the scene for the problem-solving part.

Another characteristic of the proofs produced by LEAN users is the use of words to accompany the mathematical symbols following something akin to an academic style of writing (Lew & Mejía-Ramos, 2019) in proof writing. In most cases, as it is for example in the work by Leonardo (Fig. 5), the proofs were composed of full sentences from the start, sentences which included mathematical symbols, and punctuation as well as English words. This presence of readable sentences, as opposite to strings of mathematical symbols, as we can for example see in Nataly's proof (Fig. 14), may show awareness of the role of proof as a means of communication of mathematics, as outlined for example in Hanna (1990). Regarding the idea of 'translation' (see Fig. 19 interview extract) between common language and mathematical language it is plausible to hypothesise that the exercise of communicating a proof to LEAN via writing code alerts the students of the importance of the communication aspect. Indeed, some students, as in the transcript from Luke in Fig. 21, commented on the fact that LEAN may not 'take' the proof they have written, as if the software was an external other to whom the proof had to be communicated. Even successful proofs written by students who did not engage with the LEAN workshops (see for example Nataly's proof in Figs. 14 and 17) were often difficult to read as they lacked sentence structure and linking words, as communication to others was not among the aims of the proof which was written. It also appears that LEAN users were aware of the convention of writing proofs in academic language that mathematicians see as necessary (Lew & Mejía-Ramos, 2019) while students who did not engage with LEAN were not aware of this convention.

Finally, these are all proof writing habits which are also connected to successful proofs production, especially when proofs become more complex and involve a number of distinct mathematical objects (Selden & Selden, 2008).

**Proof Structure** The second common characteristic of the proofs produced by LEAN users is the (often overt) breakdown of the proof goal in intermediate sub-goals, as it would be required for a proof to be programmed in LEAN. These sub-goals help the structuring of the proof and guide its development. The awareness of the proof sub-goals sequence has also been found to be conducive to successful proof habits and has been highlighted in previous studies, such as the one by Moore (1994), in which students admitted to not knowing what the proof required them to do.



**Use of Examples** Regarding the use of examples in proof production we observed a pattern observed before in the literature ([i.e., Aricha-Metzer and Zaslavsky (2019)]). Out of the 36 students who attempted the proof of the abundant number task, only thirteen considered examples to help them with this task. Of those students, seven were successful (i.e., scored 4 or 3 in the scoring exercise). We cannot detect any common pattern here with regard to the proof of students who attended the LEAN workshops, but we notice that the only student who attended the LEAN classes and was unsuccessful in the proof, did not use examples at any point during that section of the interview preferring to attend to the logical and mathematical language which he believed was inadequate (see Fig. 21). It is difficult to infer a general pattern from the behaviour of one student, but it could be hypothesised that for this student the emphasis on the language and symbolism distracted him from reconsidering his work and using some examples to spot a general pattern for the proof.

The data and analysis presented above suggests that engaging with a software such as LEAN can bring advantages to proof writing for students and it is plausible to link the characteristics found common to the proofs of students who used LEAN to the experience with the programming language. The precision on the use of mathematical language is likely to originate from the emphasis on precision of language that programming requires, but also in this case from the immediate feedback that the students receive when writing incorrect sentences as in Fig. 2. The students who used LEAN seemed to be aware of the differences between types of languages, such as programming language, technical language and common language and may recognise the need to follow the appropriate writing rules for each of the languages in order to communicate the proof correctly. Moreover the experience of having to ‘communicate’ the proof to LEAN may have encouraged the students to take up the habit of writing mathematics precisely and in its own academic style. These findings substantiate what Avigad (2019) also hypothesised, namely that the experience with LEAN allowed students to be well versed in the language of logic, to develop precision when handling mathematical symbols and made them aware of the difference between formal and informal proofs with special attention to the role of communication and explanation that a proof has. The setting of goals and subgoals for the proofs observed in LEAN users may be linked to the requirement of the software that proofs are organised in this way, as we can see in Fig. 1. This goal setting habit may ultimately help the students to clarify what the proof requires them to show and what steps are needed to get there.

Of course, it is possible to consider alternative explanations of the presence of the characteristics of proofs mentioned above. The students who engaged with the LEAN workshops were highly motivated and may have discussed general characteristics of proof with their peers and the Professor, although we did not observe this during the observations of the voluntary workshops. Their collaborative work on writing proof may also have made overt to them the need for communication, this time with their peers, and the need for clear writing and this may have been reflected in their proofs. Whereas we cannot be sure of what was said and done in all the workshops and outside this activity, the observations we did indicate that the focus of the workshops was to advance the LEAN library and write code for some of the proofs present in the curriculum of the module and not - at least overtly - how to write proofs in pen and

paper. This observation is also connected to some of the limitations of the study. The number of students who attended the LEAN workshops regularly was very small and therefore it is not possible to draw general conclusions beyond the indication that it is likely that engaging with the software had an impact on proof writing and proof production for those students. The second limitation of the study is that the university where the study took place is renowned worldwide for its research and it admits only students with the highest entrance requirements. It is possible to think that the results of the study may be different in a different university, or that an intervention such as one involving LEAN as a compulsory part of the curriculum may be significantly beneficial only for high achieving students. Given the data collected for this study it is not possible to discard this hypothesis. As a future direction for this research we suggest investigating the types of reasoning that students adopt when writing proofs in LEAN. This could be achieved by involving a cohort of students who have LEAN programming as compulsory part of their instruction and ascertaining whether students - at all levels of achievement - benefit from this intervention in terms of proof writing and proof production. However, the exploratory nature of the study means that the link between successful proof production and proof writing and engagement with LEAN is highly plausible and further research should aim at investigating outcomes when LEAN programming is, for example, a compulsory part of a transition to proof module.

## Concluding Remarks

The aim of this study was to ascertain whether engagement with an automated theorem prover could support proof writing and proof production habits that have been identified in the literature as successful. In order to do so we qualitatively analysed proof outcomes of students who engaged and did not engage with voluntary workshops on LEAN programming and were at the same level of achievement. The rationale behind this choice is that although we cannot claim any finding regarding achievement of correct proofs, as the sample and nature of the study does not warrant this analysis, we can however notice qualitative differences which are described as desirable by the proof writing literature. The qualitative analysis of the interview outputs did indeed show evidence of two characteristics common to LEAN users which we linked to aspects of using the programming software. These characteristics - accuracy of the use of mathematical language and proof writing resembling academic style, and division of proofs in goals and sub-goals - are positive characteristics which may in the future support successful proof production. While the sample discussed in this study is an opportunistic sample and the students who used LEAN were self-selected, we believe that the evidence presented in this paper warrants attention to the potential of the use of an automated theorem prover in an introduction to proof module as part of the curriculum. Such an implementation may have three advantages: it may help students with developing proving habits which are conducive to successful proofs, it may introduce a programming aspect to modules often taught very traditionally and it may help bridging the gap between the way in which mathematics is taught and the way in which modern mathematics evolves by allowing

students to become familiar with some of the tools used in this discipline. The latter is a gap that has been mentioned often in the literature (e.g., Artigue (2016)). Lastly, such an introduction could increase the use of programming software in university mathematics, and especially in pure mathematics modules, which is desirable in the light of the increasing importance given to programming skills in the workplace, as also suggested by the Quality Code for Higher Education for Mathematics in the UK (Lawson et al., 2015).

## Appendix 1

Below are the questions included in the questionnaire, targeting students' familiarity with LEAN.

- I have engaged with Lean (Xena Project<sup>4</sup>). [TRUE, FALSE]
- I have been to the Thursday evening sessions on: (Mark all correct responses) [A list of the dates, and “I am planning to attend future Lean (Xena project) sessions”]
- How would you characterise the following statement? I like seeing the introduction to Lean (Xena project) in our lectures. [Likert scale - Strongly Agree,...., Strongly Disagree]
- How would you characterise the following statement? When I initially engaged with the Lean materials, I found it easy to understand how it works. [Likert scale - Strongly Agree,...., Strongly Disagree]
- How would you characterise the following statement? I feel that I have benefited from using Lean in proving statements using pen and paper. [Likert scale - Strongly Agree,...., Strongly Disagree]
- I have visited the Xena Project blog.<sup>5</sup> [TRUE, FALSE]
- I have visited the Zulip<sup>6</sup> stream. [TRUE, FALSE]
- I have actively participated (asked questions, responded to comments) in the Zulip stream. [TRUE, FALSE]
- If you have engaged with Lean, are you finding this engagement useful for you? If so in what ways has this been useful for you? If you haven't engaged with Lean, please type “Not applicable” or leave empty. [open ended]
- If you have engaged with Lean, do you have any further comments you would like to add regarding your use of Lean (Xena project)? If you haven't engaged with Lean yet, please type “Not applicable” or leave empty. [open ended]
- If you haven't yet engaged with Lean (Xena project) do you think that you might in the future? Please, also provide some information as to why you would choose to engage or not. If you have already engaged with Lean, please type “Not applicable” or leave empty. [open ended]

---

<sup>4</sup>The professor had named this project as the Xena Project.

<sup>5</sup>In this blog the professor shares information about the project, recent achievements and the current state of the LEAN's library.

<sup>6</sup>Zulip is the name of the blog about LEAN that the students had access and could contribute to.

## Appendix 2

Here are the mathematical tasks the students were given during the interviews. In no interview a student was able to complete all the tasks, and the tasks were proposed to the students one at the time.

1. Prove that  $\sqrt{3} + \sqrt{5} < 4$ .
2. Let  $f : \mathbb{R} \rightarrow \mathbb{R}$  be a real-valued function. The  $f$  is called *fine* if it has a root (zero) at each integer. In other words  $f$  is fine if  $n \in \mathbb{Z} \Rightarrow f(n) = 0$ .

### Tasks:

- (a) Let  $f, g : \mathbb{R} \rightarrow \mathbb{R}$  be fine functions. Prove that  $f + g$  is a fine function. (Here  $(f + g)(x) = f(x) + g(x)$ )
  - (b)  $f : \mathbb{R} \rightarrow \mathbb{R}$  be a fine function. Let  $g : \mathbb{R} \rightarrow \mathbb{R}$  be a function defined by  $g(x) = f(x - k)$  for some  $k \in \mathbb{N}$ . Prove that  $g$  is a fine function.
3. An *abundant number* is an integer  $n$  whose divisors add up to more than  $2n$ . A *perfect number* is an integer  $n$  whose divisors add up to exactly  $2n$ .

### Tasks:

- (a) If  $n$  is perfect, then  $kn$  is abundant for any  $k \in \mathbb{N}$ .
  - (b) If  $p_1$  and  $p_2$  are primes, then  $p_1 p_2$  is abundant.
4. Let  $A \subset \mathbb{R}$  and  $f : A \rightarrow \mathbb{R}$ . The function  $f$  is *preserved in A* if  $f(A) \subset A$ .

### Tasks:

- (a) Prove that if  $f$  is strictly decreasing function, then there is a fine set  $A$  such that  $f$  is preserved in  $A$ .
- (b) Let  $f, g : \mathbb{R} \rightarrow \mathbb{R}$  be functions and  $a \subset \mathbb{R}$ . Suppose  $x \times y \in A$  for all  $x, y \in A$ . Prove that if  $f$  and  $g$  are preserved in  $A$  then  $f \times g$  is preserved in  $A$ . (Here  $f \times g = f(x) \times g(x)$ .)

**Acknowledgements** We would like to thank Professor Kevin Buzzard and the students who participated in our study. We would also like to thank Adrian Simpson for the help with the statistical analysis and Apostolos Thoma who assisted in the independent scoring of the proofs.

**Funding** We would like to thank Imperial College London for funding this research through the Pedagogy Transformation Fund.

## Declarations

**Conflict of Interests** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Aricha-Metzer, I., & Zaslavsky, O. (2019). The nature of students' productive and non-productive example-use for proving. *The Journal of Mathematical Behavior*, 53, 304–322.
- Artigue, M. (2016). Mathematics education research at university level: Achievements and challenges. In *Proc. 1st Conf. of INDRUM* (pp. 11–27).
- Avigad, J. (2019). Learning logic and proof with an interactive theorem prover. In *Proof technology in mathematics research and teaching* (pp. 277–290). Springer.
- Azrou, N., & Khelladi, A. (2019). Why do students write poor proof texts? a case study on undergraduates' proof writing. *Educational Studies in Mathematics*, 102(2), 257–274.
- Broda, K., Ma, J., Sinnadurai, G., & Summers, A. (2007). Pandora: a reasoning toolbox using natural deduction style. *Logic Journal of the IGPL*, 15(4), 293–304.
- Broley, L. (2016). The place of computer programming in (undergraduate) mathematical practices. In *Proceedings of INDRUM 2016 first conference of the international network for the didactic research in university mathematics* (pp. 360–369).
- Broley, L., Caron, F., & Saint-Aubin, Y. (2018). Levels of programming in mathematical research and university mathematics education. *International Journal of Research in Undergraduate Mathematics Education*, 4(1), 38–55.
- Buteau, C., Gueudet, G., Muller, E., Mgombelo, J., & Sacristán, A. I. (2020). University students turning computer programming into an instrument for 'authentic' mathematical work. *International Journal of Mathematical Education in Science and Technology*, 51(7), 1020–1041.
- Cornu, B. (1991). Limits. *Advanced Mathematical Thinking*, pp 153–166.
- Dawkins, P. C., & Weber, K. (2017). Values and norms of proof for mathematicians and students. *Educational Studies in Mathematics*, 95(2), 123–142.
- Dreyfus, T. (1999). Why Johnny can't prove. In *Forms of mathematical knowledge* (pp. 85–109). Springer.
- Dubinsky, E. (1984). The cognitive effect of computer experiences on learning abstract mathematical concepts. *Korkeakoulujen Aik-Vuuset*, 2, 41–47.
- Dubinsky, E. (1995). Isetl: a programming language for learning mathematics. *Communications on Pure and Applied Mathematics*, 48(9), 1027–1051.
- Dubinsky, E., & Leron, U. (1994). *Learning abstract algebra with ISETL*. Springer Science & Business Media.
- Ehle, A., Hundeshagen, N., & Lange, M. (2018). The sequent calculus trainer with automated reasoning-helping students to find proofs. arXiv:1803.01467.
- Fukawa-Connelly, T. (2012). Classroom sociomathematical norms for proof presentation in undergraduate in abstract algebra. *The Journal of Mathematical Behavior*, 31(3), 401–416.
- Gillham, B. (2005). *Research interviewing: The range of techniques: A practical guide*. UK: McGraw-Hill Education.
- Hanna, G. (1990). Some pedagogical aspects of proof. *Interchange*, 21(1), 6–13.
- Harel, G., & Sowder, L. (1998). Students' proof schemes: Results from exploratory studies. *American Mathematical Society*, 7, 234–283.
- Harel, G., & Sowder, L. (2007). Toward comprehensive perspectives on the learning and teaching of proof. *Second handbook of research on mathematics teaching and learning*, 2, 805–842.
- Lawson, D., Arrowsmith, D., & Bailey, T. (2015). *Mathematics statistics and operational research*. Gloucester: The Quality Assurance Agency for Higher Education.
- Lee, K., & Smith, J. P. III. (2009). Cognitive and linguistic challenges in understanding proving. In *Proceedings of the ICMI Study 19 Conference: Proof and Proving in Mathematics Education*, (Vol. 2 pp. 21–26).
- Leron, U. (1983). Structuring mathematical proofs. *The American Mathematical Monthly*, 90(3), 174–185.
- Lew, K., & Mejía-Ramos, J. P. (2019). Linguistic conventions of mathematical proof writing at the undergraduate level: Mathematicians' and students' perspectives. *Journal for Research in Mathematics Education*, 50(2), 121–155.
- Lo, C. K., Hew, K. F., & Chen, G. (2017). Toward a set of design principles for mathematics flipped classrooms: a synthesis of research in mathematics education. *Educational Research Review*, 22, 50–73.
- Mejía-Ramos, J. P., Fuller, E., Weber, K., Rhoads, K., & Samkoff, A. (2012). An assessment model for proof comprehension in undergraduate mathematics. *Educational Studies in Mathematics*, 79(1), 3–18.

- Moore, R. C. (1994). Making the transition to formal proof. *Educational Studies in mathematics*, 27(3), 249–266.
- Pesonen, M. E., & Malvela, T. (2000). A reform in undergraduate mathematics curriculum: More emphasis on social and pedagogical skills. *International Journal of Mathematical Education in Science and Technology*, 31(1), 113–124.
- Raman, M. (2003). Key ideas: What are they and how can they help us understand how people view proof? *Educational studies in mathematics*, 52(3), 319–325.
- Rasmussen, C., & Kwon, O. N. (2007). An inquiry-oriented approach to undergraduate mathematics. *The Journal of Mathematical Behavior*, 26(3), 189–194.
- Sangwin, C. J., & O’Toole, C. (2017). Computer programming in the uk undergraduate mathematics curriculum. *International Journal of Mathematical Education in Science and Technology*, 48(8), 1133–1152.
- Selden, A. (2011). Transitions and proof and proving at tertiary level. In *Proof and proving in mathematics education* (pp. 391–420). Springer.
- Selden, A., & Selden, J. (2007). Teaching proving by coordinating aspects of proofs with students’ abilities. technical report. no. 2007-2. Online Submission.
- Selden, A., & Selden, J. (2008). Overcoming students’ difficulties in learning to understand and construct proofs. Making the connection: Research and teaching in undergraduate mathematics, pp 95–110.
- Talbert, R. (2015). Inverting the transition-to-proof classroom. *Primus*, 25(8), 614–626.
- Weber, K. (2004). Traditional instruction in advanced mathematics courses: a case study of one professor’s lectures and proofs in an introductory real analysis course. *The Journal of Mathematical Behavior*, 23(2), 115–133.
- Weber, K., & Alcock, L. (2004). Semantic and syntactic proof productions. *Educational studies in mathematics*, 56(2-3), 209–234.
- Zazkis, D., Weber, K., & Mejía-Ramos, J. P. (2015). Two proving strategies of highly successful mathematics majors. *The Journal of Mathematical Behavior*, 39, 11–27.

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.