A Technological Approach to Teaching Inequalities, Propositional and Predicate Logic

Zoltán Kovács[1,*,†], Reinhard Oldenburg[2,†]

[1] The Private University College of Education of the Diocese of Linz, Salesianumweg 3, Linz, 4020, Austria
[2] Augsburg University, Institute of Mathematics, Universitätsstraße 2, Augsburg, 66159, Germany

Abstract

A conceptual understanding of inequalities and logic is essential for success in most of mathematics and its applications. E.g., it is indispensable for the learning of calculus because inequalities and logic are the tools that are used to express intuitive ideas about rates, changes and limits in a formal way that is required to give proofs. We show that GeoGebra and especially GeoGebra Discovery can be used to support the learning of these foundations. It will be shown that propositional logic is already supported well in GeoGebra and that support for predicate logic is realized in the experimental fork GeoGebra Discovery.

Keywords

Quantifier elimination, Calculus education, Inequalities, Logic

1. Introduction

Students face a lot of obstacles when starting to learn advanced mathematics and especially calculus [2]. Among these obstacles is the understanding of inequalities and the ability to manipulate them, the role of logic and especially quantifiers (see e.g. [17, 19, 20, 18]). However, more in-depth studies that investigate how the learning process of logic can be supported seem to be absent. Durand-Guerrier and Dawkins [7] have noted that "Part of the study of logic in undergraduate mathematics education should include students' learning of such formal systems in model theory, proof theory, or computability. However, we are aware of no research-based evidence on the teaching and learning of such topics." The present paper does not fill this gap, either, but it discusses technology that might offer learning opportunities in the domain of logic.

Building up competencies in logic and logical reasoning by mere paper & pencil exercises is difficult because misconceptions may prevail for a long time. When students work, however, with digital tools they get immediate feedback and can extend the range of examples beyond what is possible by time-consuming calculations. In this paper, we show what activities are possible in GeoGebra and GeoGebra Discovery to support learning.

Possible ways to use computer algebra in the classroom has been a widely studied topic of scientific communities in the last decades. Being an important snapshot on post-secondary education from the early 2010s, we refer to [14]. To mention another important milestone, we recall that in 2014, a team of mathematics teachers and free software developers decided to implement a symbolic computation suite for the popular dynamic mathematics software GeoGebra. The initial attempts covered the high school final mathematics exam in Bavarian schools in 2014 (see a collection of problems and their solutions with GeoGebra at https://www.geogebra.org/m/cCUCGSxN). After the first successful results, in Austria and Germany,

9th International Workshop on Satisfiability Checking and Symbolic Computation, July 2, 2024, Nancy, France, Collocated with IJCAR 2024

*Corresponding author.
†These authors contributed equally.
EMAIL: zoltan@geogebra.org (Z. Kovács); reinhard.oldenburg@math.uni-augsburg.de (R. Oldenburg)
URL: https://matek.hu/zoltan (Z. Kovács)
ORCID: 0000-0003-2512-5793 (Z. Kovács); 0000-0002-3870-9248 (R. Oldenburg)

GeoGebra was officially designated as preferred tool for learners, because of its symbolic computation capabilities, easy syntax and free availability. As of today, various school books include tutorials and explanatory exercises on solving problems with the help of GeoGebra's CAS Window (see, for example [9]).

Accessing symbolic algorithms in most classrooms is, however, still limited to widely-known techniques only. It is quite well-accepted by teachers that a computer algebra system should be able to solve a set of linear equations, or to find the derivative or anti-derivative of a function. But it is quite surprising for many teachers that today's algorithms are ready to automatically solve problems that contain quantifiers as well. In fact, many teachers still lack good knowledge of connecting quantified formulas and text problems. Certainly, stable computer algebra support to handle such formulas with an easy syntax and a robust backend would speed up the widespread of the required knowledge among teachers and learners as well.

Therefore, our contribution confirms the relevance of the SC^2 Project for a wider community, including schools. The new cylindrical algebraic decomposition (CAD) algorithms that came out of SC^2 Project (for example, the non-uniform CAD algorithm in TARSKI) already make it possible to bring robust computations of non-trivial classroom problems by exploiting the popular user interface of GeoGebra. On the other hand, there is still room for several improvements. We give some examples on possible future work in the last section of the paper.

2. Didactical Background

Logical thinking and mathematics are often believed to be related, and this is supported by several studies. Wong [23] finds correlations between deductive reasoning and mathematical problem-solving. Cresswell and Speelman [6] found a significant correlation (although only 0.135) between logical thinking and scientific achievement.

One of the main issues in learning propositional logic is getting used to the definition of implication, which is not meant to represent transfer of epistemic support, but only the formal relation between truth values. Related to this is the law of excluded middle or equivalently that double negation on a statement is logically identical to the original statement (of course, it should be noted that these issues are deep and that there are constructivist mathematicians that do not accept proofs by contradiction, but for the present paper, we simply restrict ourselves to classical logic). Durand-Guerrier and Dawkins [7] mention that therefore the use of natural language in logic education has some benefits, but that "Coordinating natural and mathematical language requires striking a careful balance." We argue that using the logical operators in connection with properties of points in the Euclidean plane provides an environment that is concrete enough to let students easily get the meaning of propositions, yet abstract enough to avoid pitfalls of natural language.

Propositional logic is not enough for mathematics: only predicate logic gives the means to deal with mathematical objects like numbers and functions and describe their properties in a precise sense. This role of describing properties precisely and enabling correct argumentation about them is also important beyond mathematics, e.g. in computer science. The book [10] expresses this clearly in its subtitle "Modelling and Reasoning about Systems".

Quantifiers play an essential role in predicate logic, yet they are difficult for learners to understand correctly (see e.g. [17, 19]). In many situations in mathematics, their use can be avoided by expressing them in common language. Hence, some students view formalization as an additional burden, not as a useful concept. In a computer algebra environment, however, formalization is the way to communicate with the computer and thus this can give students a feeling for its sense. Moreover, quantifiers are essential in the definitions of many concepts, especially in rigorous calculus. Formalization can thus be seen as a modelling process: intuitive concepts are modelled in formalized language, and a computer environment can be used to validate the results. This point of view, that formalization is a kind of modelling, has been advocated in the paper [15] (Fig. 2), but shall be elaborated with another example: students easily have an intuitive understanding of what it means that a function on an interval $[a, b]$ increases strictly, e.g. they might express this as the rule that "if x increases, so does $f(x)$". However, to give proofs a formal model of this intuitive idea must be developed, namely

$$\forall x_1, x_2 \in [a, b] : x_1 < x_2 \Rightarrow f(x_1) \leq f(x_2).$$

Thus, we will show that technology can be used to exercise such formalisation processes. Dual to formalization is interpretation, e.g. giving meaning to a formula. Both activities are linked in the modelling cycle and both activities should be addressed in teaching.

3. Mathematical Background

Tarski [21] has shown that in a certain class of ordered fields, including the real numbers, the predicate logic that includes addition, multiplication, equality (=), and inequality (<), is decidable. Moreover, he showed that every formula of this language, including quantifiers, can be transformed algorithmically into an equivalent form that is free of quantifiers. The theory covers absolute values, polynomials (and with some care also rational functions) as well as powers to fixed rational exponents, but no logarithms, exponentials or other transcendental functions.

There are different realizations of Tarski's theory. The most important one is based on CAD. Metaphorically speaking, it rests on dividing the space of possible values of variables into components, on which each polynomial relevant in the formula at hand has the same sign. The question as to whether a formula is true for all infinitely many numbers can thus be reduced to the question if it is true on a finite set of sample point: one from each of these cells. Although explaining CAD and how it leads to quantifier elimination is difficult (an attempt has been made in [1]) it is nevertheless sensible to relate it to the ways algebraic inequalities can be used to segment the Euclidean plane.

4. Technical Background

Recently, an experimental version of *GeoGebra* [8], *GeoGebra Discovery* [5] started to support computations via CAD. An underlying system called TARSKI [22] has been embedded in GeoGebra Discovery via a Java Native Interface (for the desktop version) and a JavaScript library (for the web version, available at https://autgeo.online). In fact, TARSKI forwards the input problems in a prepared and rewritten form for the underlying QEPCAD B [3] system, which is a well-known industry standard software in solving problems that require quantifier elimination.

As a result, it is easy to formulate statements given in predicate logic in GeoGebra Discovery. This usually means that a combination of existential and universal quantifiers can be given, that quantify an expression that consists of polynomial equations or inequalities, connected by logical operations. For example, to formalize the statement that there is no least positive number, we can use GeoGebra Discovery's CAS Window and type the following line

$$\neg(\exists x\ x > 0 \wedge (\forall y\ y > 0 \rightarrow x < y)).$$

Now, GeoGebra Discovery preprocesses this input for TARSKI, which implies a certain extent of automatic simplification and the formula

$$\forall x\ (\exists y\ (x \leq 0 \vee y > 0 \wedge y - x \leq 0)) \tag{1}$$

is returned: it is a prenex form of the input that contains no implication operations. Finally, the command `RealQuantifierElimination($)` computes via a black-box CAD process the quantifier free version which is in this case simply the logical truth of the formula, i.e. it returns the answer "true".

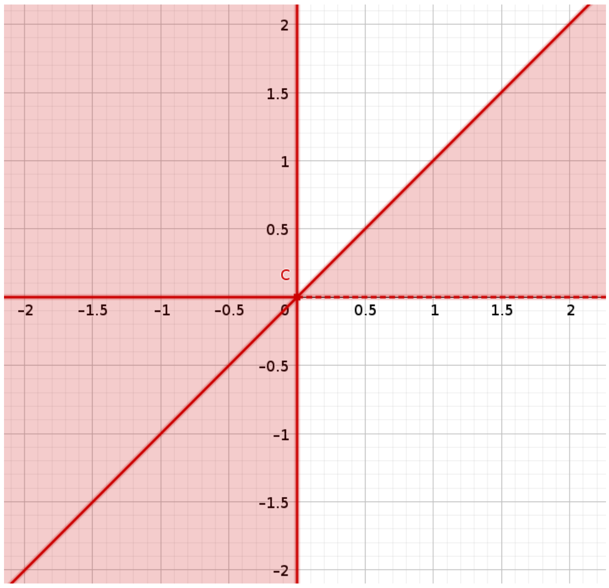

Figure 1: A CAD with respect to variable x.

GeoGebra Discovery also supports visualizing CAD in two dimensions. Here, the command `Plot2D(`$x \leq 0 \vee y > 0 \wedge y - x \leq 0$`, "x")` shows the decomposition

$$(A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5) \cup (B_1 \cup B_2 \cup B_3) \cup (C_1 \cup C_2 \cup C_3 \cup C_4 \cup C_5)$$

where $A_1 = \{x < 0 \wedge y < x\}$, $A_2 = \{x < 0 \wedge y = x\}$, $A_3 = \{x < 0 \wedge x < y < 0\}$, $A_4 = \{x < 0 \wedge y = 0\}$, $A_5 = \{x < 0 \wedge 0 < y\}$, $B_1 = \{x = 0 \wedge y < 0\}$, $B_2 = \{x = 0 \wedge y = 0\}$, $B_3 = \{x = 0 \wedge y > 0\}$, $C_1 = \{0 < x \wedge y < 0\}$, $C_2 = \{0 < x \wedge y = 0\}$, $C_3 = \{0 < x \wedge y < x\}$, $C_4 = \{0 < x \wedge y = x\}$, $C_5 = \{0 < x \wedge x < y\}$ (see Figure 1). One can easily see that the projections of these sets on the x-axes are $A = \{x < 0\}$, $B = \{x = 0\}$ and $C = \{0 < x\}$, respectively. On the other hand, it is easy to read off that formula (1) is true because for all projected sets A, B and C, there are pre-image regions (namely, A_1, A_2, A_3, A_4, A_5, B_1, B_2, B_3, C_3 and C_4) such that the formula $x \leq 0 \vee y > 0 \wedge y - x \leq 0$ yields.

In fact, the CAD approach allows the user to plot arbitrary inequalities or logical connectives of inequalities in two variables in a faithful way. This can be, however, computationally exhausting for formulas containing many logical operations or higher degrees of polynomials [4].

5. Propositional Calculus and Inequalities

With this section, we start with presenting ideas how these tools can be included in teaching. Inequalities in two variables have the nice property that their solution set can be represented graphically, and standard GeoGebra can do this easily (without using a CAD, unless higher degree polynomials are present). Moreover, it offers the possibility to combine inequalities by logical junctions such as “and”, “or”, “not”, and “implication”. For example, one may enter $y > 0 \wedge x^2 + y^2 < 9$ to get directly the graph shown in Figure 2 on the left. To get the result on the right side of Figure 2, students can come up with several solutions. What they do is modelling geometric sets within logic and set theory. Experience showed (see [16]) that such activities can be motivating even for younger students of age 15. The graphical context eased argumentation and students easily acquired logical rules such as de Morgans’ laws, the rule of double negation or the expression of implication by conjunction $a \rightarrow b \Leftrightarrow \neg a \vee b$. Such activities can also be combined with absolute values.

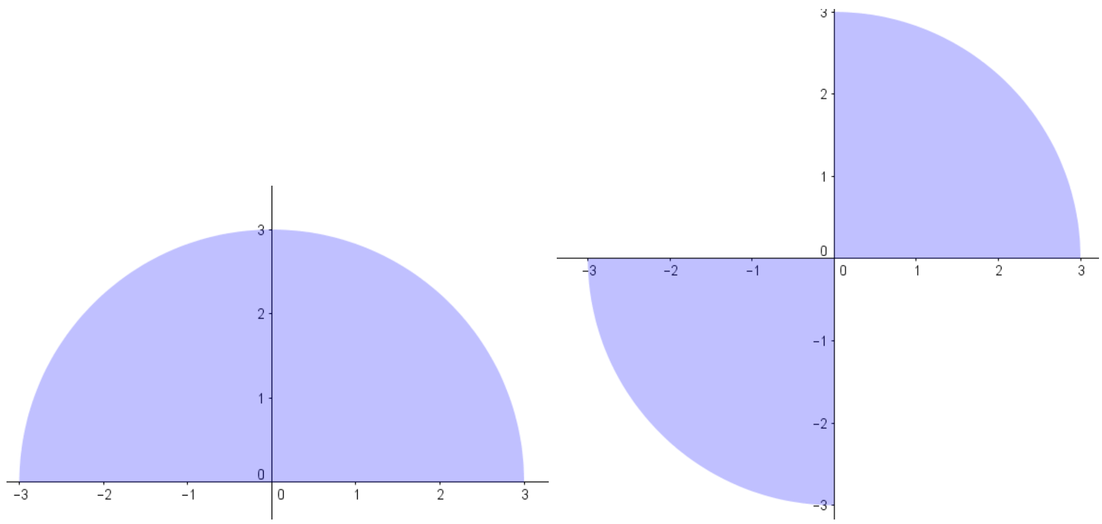

Figure 2: Some regions drawn by GeoGebra from logical descriptions.

6. Predicate Calculus: Handling Quantifiers in Calculus

Formalization of intuitive ideas with predicate logic and understanding formulas expressed in predicate logic are dual activities that can both be exercised within GeoGebra Discovery. Let us first consider two easy examples taken from assignments in actual teaching, one of each kind:

1. Interpretation: Given the real function $f(x) = x^3 - x$ interpret what the following formula means: $\forall x : x > a \to (\forall y : x < y \to f(x) < f(y))$.
2. Modelling: Express that a real valued function has exactly one, resp. exactly two global maxima. Test your formalization with various polynomial functions.

The meaning of the formula in 1 is best understood when doing some rearrangement to come to the equivalent form $\forall x, y : a < x \land x < y \to f(x) < f(y)$. Hence, the meaning is that the function is strictly monotonically increasing on the open interval $]a, \infty[$. For $f(x) = x^3 - x$ quantifier elimination will hence give the equivalent form $a \geq 1/\sqrt{3}$ which can be used to check logical reasoning.

The second example leads to the formalizations $\exists x_0\ \forall x : x \neq x_0 \to f(x) < f(x_0)$ resp. $\exists x_1\ \exists x_2 : x_1 \neq x_2 \land f(x_1) = f(x_2) \land \forall x : f(x) \leq f(x_1) \land (f(x) = f(x_1) \to (x = x_1 \lor x = x_2))$

Figure 3 shows how the first activity can be supported with GeoGebra Discovery: In its CAS View the function f must be defined first, then the quantified statement is to be entered (which will be automatically rewritten into an equivalent prenex form), after this step a real quantifier elimination is performed, which is finally reformulated into a formula which is familiar for students.

Figure 4 gives an overview on how the second part of activity 2 can be formulated in the same software. First, we define a quartic function that fulfills the expected property. Later, we should do a counter-test with a non-example function. Syntactically, it is important that in certain parts of the input a minimal number of spaces and parentheses must be used; otherwise, the input will be misinterpreted and the user can obtain, at first sight, strange results. Here we provide a "didactic trick" to obtain some analysis on two simplified formulas that contain just two quantifiers (or just one) with the same inner formula: clearly, GeoGebra cell 5 finds the extrema at for the input function, and this is confirmed in cell 7 of the session.

It is quite clear that more complicated notions and concepts may be even more difficult to formulate and check. But, unlike other computer software that support real quantifier elimination (like Mathematica, which may be a problematic choice for students because of its commercial licensing; or like QEPCAD B, which is difficult to access for non-experts), GeoGebra Discovery

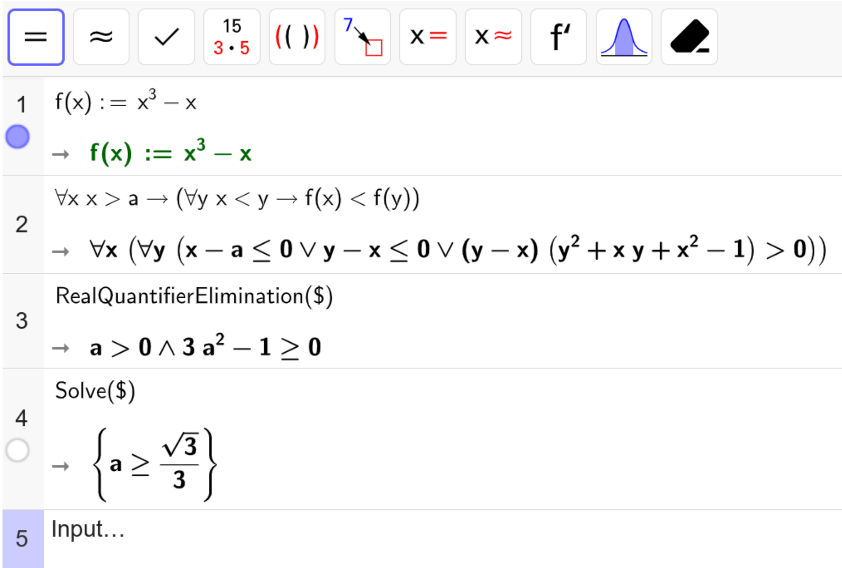

Figure 3: Computer algebra support for the first activity in GeoGebra Discovery's web version.

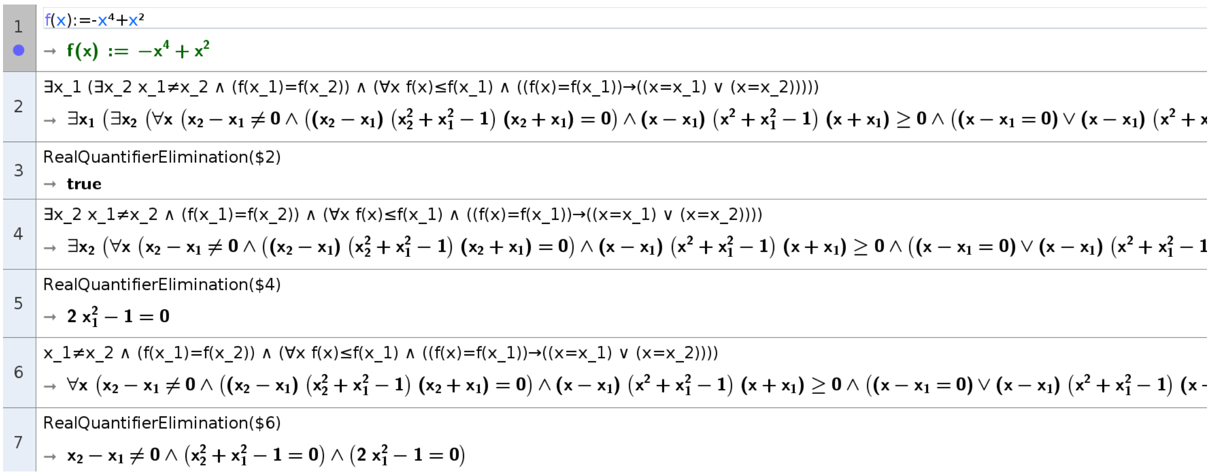

Figure 4: A CAS session in GeoGebra Discovery's desktop version, illustrating the second activity.

offers a classroom-like syntax inside its CAS View which should already be quite well-known for many learners.

To finish, we give two advanced examples to check continuity and differentiability.

First, we consider the function $f(x) = x^2/x$ if $x \neq 0$. This has the limit 0 near 0, but, since it is undefined at 0, it is not continuous. A correct formula that checks the limit property is now

$$\forall \varepsilon \; \varepsilon > 0 \to (\exists \delta \; \delta > 0 \wedge (\forall x \; x \neq 0 \wedge x < \delta \to x^2/x < \varepsilon)).$$

Here we note that the software rewrites the input into the non-equivalent form

$$\forall \varepsilon \; (\exists \delta \; (\forall x \; (\varepsilon \leq 0 \vee \delta > 0 \wedge ((x = 0) \vee (x^2(\varepsilon - x)) > 0 \vee \delta - x \leq 0))))$$

to prepare it to be applicable for Tarski's theory because divisions are disallowed. The trick here is to multiply by the square of the denominator, however, this should be done cautiously as being considered also in the original manual input. Finally, the real quantifier elimination confirms that f has a removable discontinuity at 0.

Second, we would like to compute the derivative of a rational function. In Figure 5 we collected the required steps for a simple input: a rational function of two linear formulas. Here we note

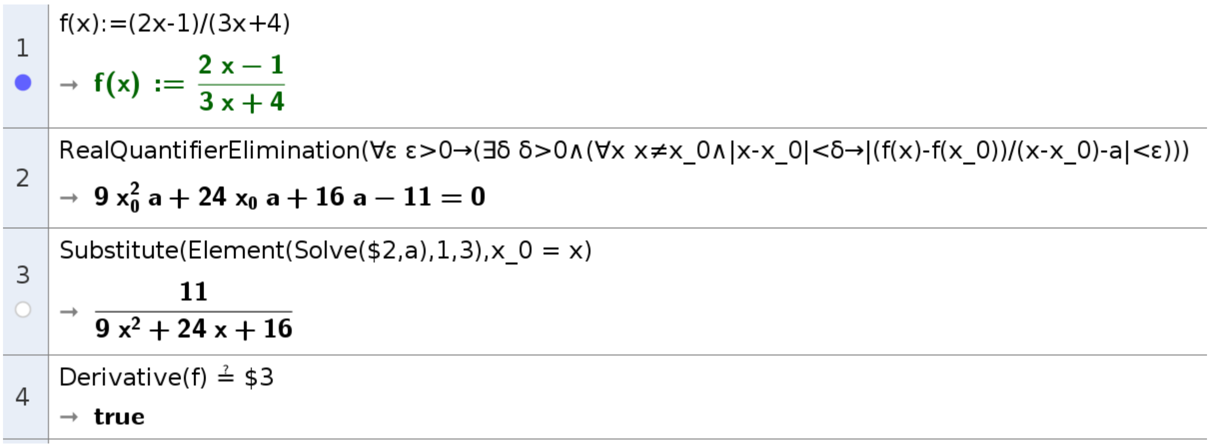

Figure 5: A CAS session confirms that the formal definition of the derivative is correct.

that it may take quite a long time until the learners find a syntactically correct input formula. So, very importantly, proper use of implications (after universal quantifiers) and conjunctions (after existential quantifiers) cannot be avoided, that is, finding a successful solution requires solving mathematical challenges as well.

7. Report on Use Cases in Teacher Training

First author held a teacher training course in logic for prospective mathematics teachers in the winter semester of 2023/2024 at The Private University College of Education of Linz, Austria. This course was a refined version of two former courses, based on a similar content but now with improvements. The lecture notes for this course are available at [11]. In particular, chapter 4.4 on real quantifier elimination (p. 51–53) focuses on secondary school applications. Also, chapter 4.8 on CAD (p. 65–69) includes various figures that were created with GeoGebra Discovery's `Plot2D` command, contributed by one of the students as an extra assignment.

The students were asked to try both commands `RealQuantifierElimination` and `Plot2D` in the desktop or web version of GeoGebra Discovery, during the lectures and also for their homework assignments. Figure 6 shows the expected homework activities for the participants, 30 students, many of them having a bachelor degree already, but some of them not yet (only successfully passed exams in discrete mathematics, linear algebra and calculus). The concept of a CAD was difficult to understand for many students, this is why the lecturer decided to prolong explaining the topic, since in the planned schedule there was only two weeks reserved for these activities.

8. Feature List

- Quantified formulas can be stored in variables.
- Formula negation is supported.
- Disjunction, conjuction and implication between two formulas is allowed.
- Output expressions are automatically converted into prenex format.
- Quantifiers ∀ and ∃ are available in GeoGebra's virtual keyboards. Also, keyboard shortcuts (Alt-V and Alt-X) are supported.
- Keyboard shortcuts for the negation sign (Alt-Z), disjunction (Alt-J), conjunction (Alt-K), implication (Alt-Y) and the not-equal sign (Alt-H).
- Localization for the `Plot2D` and `RealQuantifierElimination` commands are available in English (default), German, Spanish and Hungarian.

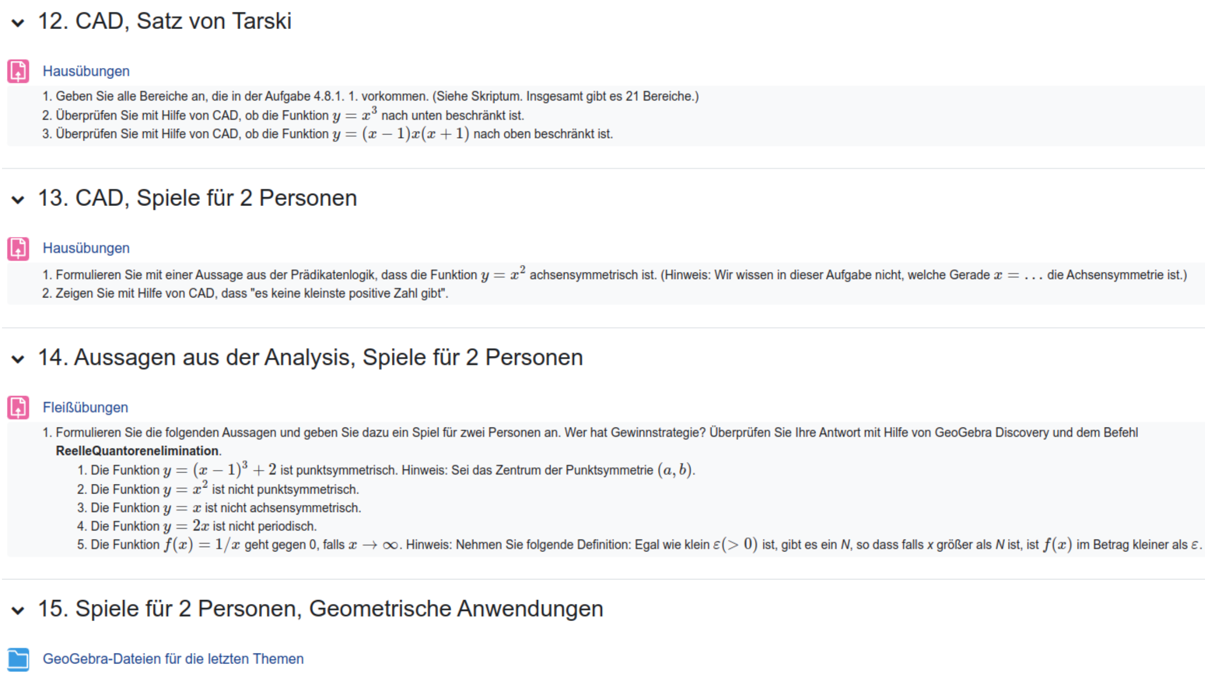

Figure 6: Homework assignments in Moodle for prospective mathematics teachers.

- Rational numbers and quadratic surds are accepted as input.
- Rational functions are rewritten as implications before conversion to TARSKI.
- The `Plot2D` command can have a second argument: "x" or "y". If it is set, the cylindrical algebraic decomposition will be created with respect to the given variable.

9. Conclusion

The above presentation showed that nowadays the technique of quantifier elimination is in the reach of students and can be used to do sensible exercises in calculus. This use can build naturally on former use of GeoGebra to exercise modelling with propositional calculus and inequalities. In all applications the combination of two dual processes, the interpretation of formulas and formalization of intuitive ideas, is an important didactical method that allows to take benefits from the technical power of GeoGebra Discovery. A systematic empirical evaluation of these benefits is, however, still to be conducted.

9.1. Connection with the SC^2 Project

Teaching the concept of proofs in classrooms with the help of technology requires reliable implementations of the used prover subsystems. In GeoGebra Discovery we use the TARSKI system for outsourcing heavy computations that require reliable and possibly fast calculations. Also, some basic operations like parsing the input or preparing it for further operations inside TARSKI, may require additional internal programmatic techniques. Here we mention the `smtlib` interface between them: for instance, the formula

```
(qepcad-api-call (smtlib-load 'clear "(assert (exists ((x Real)) (and (not (=
             a 0) ) (= (+ (+ (* x b) c) (* (* x x) a)) 0))))") 'T)
```

describes the user input *existence of a real root of the quadratic equation* $ax^2 + bx + c = 0$. After performing quantifier elimination, the result `a /= 0 /\ 4 a*c - b^2 <= 0` (which means: *the* *discriminant* $b^2 - 4ac$ *is non-negative*) must be translated back to GeoGebra internals. See [12], slide 11, for some more details.

Now, we point to some possible future work. The example in Fig. 5 performs reliably if the input f is a rational function, unless the denominator is of higher degree than 1. In other cases the computation may result in a program crash due to possible lack of resources, missing optimizations in the underlying algorithms or other issues. For a reliable student experience this should be addressed in a next version of the program.

Acknowledgments

The first author was partially supported by the project "Mathematical Visualization: Foundations, Algorithms and Applications", PID2020-113192GB-I00 from the Spanish MICINN. We thank Chris Brown for his continuous support in help using Tarski in the appropriate way for our project, and for updating it accordingly.

References

[1] Artigue, M., Oldenburg, R.: How to get rid of quantifiers. https://blog.kleinproject.org/?p=2466 (2014), last accessed 2024/05/08

[2] Bressoud, D., Ghedamsi, I., Martinez-Luaces, V., Törner, G.: Teaching and Learning of Calculus. Springer (2016). doi:10.1007/978-3-319-32975-8

[3] Brown, C.W.: Qepcad B: a program for computing with semi-algebraic sets using CADs. ACM SIGSAM Bull. **37**(4), 97–108 (2003). doi:10.1145/968708.968710

[4] Brown, C., Kovács, Z., Recio, T.: Faithful Real-Time Animation of Parametrized (Semi-) Algebraic Expressions via Cylindrical Algebraic Decomposition. ACM Commun. Comput. Algebra 57.2, 43–46 (2023). doi:10.1145/3614408.3614413

[5] Brown, C.W., Kovács, Z., Recio, T., Vajda, R., Vélez, M.P.: GeoGebra Discovery (system entry). In: Dubois, C., Kerber, M. (eds.) Intelligent Computer Mathematics. 16th International Conference, CICM 2023, LNCS, vol. 14101, pp. 324. Springer, Cham (2023). doi:10.1007/978-3-031-42753-4

[6] Cresswell, C., Speelman, C.P.: Does mathematics training lead to better logical thinking and reasoning? A cross-sectional assessment from students to professors. PLoS ONE **15**(7), e0236153 (2020). doi:10.1371/journal.pone.0236153

[7] Durand-Guerrier, V., Dawkins, P.C.: Logic in University Mathematics Education. In: Lerman, S. (eds.) Encyclopedia of Mathematics Education. Springer (2018). doi:10.1007/978-3-319-77487-9_100024-1

[8] Hohenwarter, M.: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene, Master's thesis, Paris Lodron University, Salzburg (2002)

[9] Humenberger, H., Hasibeder, J., Himmelsbach, M., Schüller-Reichl, J., Litschauer, D., Groß, H., Aue, V.: Das ist Mathematik 3. Österreichischer Bundesverlag Schulbuch, Wien (2019).

[10] Huth, M., Ryan, M.: Logic in Computer Science. Modelling and Reasoning about Systems. Cambridge (2004).

[11] Kovács, Z.: Logik für Lehramt Mathematik. https://github.com/kovzol/logik/releases/tag/April_2024 (2024). Last accessed 2024/06/13

[12] Kovács, Z., Éliás, J.: A derivált epszilon-deltás formális definíciójáról – számítógéppel. NJ80: Szimpózium az elmúlt évek analízis-tanítási módszereiről (2023). doi:10.13140/RG.2.2.15097.44646, last accessed 2024/05/08

[13] Kovács, Z., Recio, T.: Real Quantifier Elimination in the Classroom. In: Yang, W.C., Majewski, M., Meade, D., Ho, W.K. (eds.) Electronic Proceedings of the 27th Asian Technology Conference in Mathematics. (2022). https://atcm.mathandtech.org/EP2022/invited/21952.pdf, last accessed 2024/05/08

[14] Buteau, C., Marshall, N., Jarvis, D.H., Lavicza, Z.: Integrating computer algebra systems in post-secondary mathematics education: Preliminary results of a literature review. International Journal for Technology in Mathematics Education **16**(2) (2010).

[15] Oldenburg, R.: Gains and pitfalls of quantifier elimination as a teaching tool. The International Journal for Technology in Mathematics Education **22**, 163–167 (2015) Retrieved June 5, 2024 from https://www.learntechlib.org/p/195004/.

[16] Oldenburg, R.: Logik und Ungleichheit – ein leider exotisches Thema. In: Bender, P. (ed.) Beiträge zum Mathematikunterricht 2018, pp. 1347–1350, WTM-Verlag (2018)

[17] Selden, J., Selden, A.: Unpacking the logic of mathematical statements. Educational Studies in Mathematics **29**, 123–151 (1995) doi:10.1007/BF01274210

[18] Selden, A.: Transitions and Proof and Proving at Tertiary Level. In: Hanna, G., de Villiers, M. (eds.) Proof and Proving in Mathematics Education. Springer (2012). doi:10.1007/978-94-007-2129-6_17

[19] Shipman, B.: Subtleties of hidden quantifiers in implication. Teach. Math. Appl. **35**(1), 41–49 (2016). doi:10.1093/teamat/hrv007

[20] Stavrou, S.: Common errors and misconceptions in mathematical proving by education undergraduates. Issues Undergrad. Math. Prep. Sch. Teach., 1–8 (2014)

[21] Tarski, A.: A decision method for elementary algebra and geometry. Rand Corporation Publication (1948). doi:10.2307/2267068

[22] Vale-Enriquez, F., Brown, C.W.: Polynomial Constraints and Unsat Cores in TARSKI. In: Davenport, J. H., Kauers, M., Labahn, M., Urban, J. (eds.) Mathematical Software – ICMS 2018, LNCS, vol. 10931, pp. 466–474. (2018). doi:10.1007/978-3-319-96418-8_55

[23] Wong, T.: Is conditional reasoning related to mathematical problem solving? Developmental Science **21**(5), e12644 (2017). doi:10.1111/desc.12644