

# Using Automated Reasoning Tools to Explore Geometric Statements and Conjectures



Markus Hohenwarter, Zoltán Kovács and Tomás Recio

## 1 Introduction

*GeoGebra*, a software tool for dynamic mathematics, has recently been enhanced with an automated reasoning subsystem able to confirm/deny the truth of any geometry statement displayed on the screen. In addition, if the statement is labeled as *false*, *GeoGebra* can exhibit required modifications to the hypotheses that make the statement true. The free availability and portability of *GeoGebra* have made it possible for millions of students worldwide to harness these novel techniques on tablets, smartphones, and computers.

The mathematical background of this reasoning method—based on automatically algebraizing a given geometric statement and associated construction and then applying effective algebraic geometry tools—goes back to the work of Wen-Tsün Wu in the 1970s (see Wu, 1978; Chou, 1987). Wu's highly performing approach was the starting point for the developing and implementing different algebraic geometry based, automated reasoning algorithms in a large collection of programs. However, there has never been a program as effective as *GeoGebra* in:

- merging dynamic geometry and computer algebra,
- addressing non-experts, and
- achieving an impact in the educational community worldwide.

---

M. Hohenwarter  
Johannes Kepler University of Linz, Linz, Austria  
e-mail: [markus.hohenwarter@jku.at](mailto:markus.hohenwarter@jku.at)

Z. Kovács (✉)  
The Private University College of Education of the Diocese of Linz, Linz, Austria  
e-mail: [zoltan@geogebra.org](mailto:zoltan@geogebra.org)

T. Recio  
University of Cantabria, Santander, Spain  
e-mail: [tomas.recio@unican.es](mailto:tomas.recio@unican.es)

In this chapter we propose the exploration of a new kind of workflow in the teaching/learning of elementary geometry with the help of the GeoGebra Automated Reasoning Tool (ART). Let us remark that the very recent launching of this tool makes our work a proposal not yet supported by experience. However, as we summarize below and argue in more detail in the conclusion, we think it is important to reflect on the potential uses and misuses of these tools in education early on.

It is well known that dynamic geometry systems (DGS), even without these specific features, can be useful (as well as challenging) tools in the teaching/learning of reasoning and proof. DGS allow students to formulate certain geometric facts (e.g., as intermediate steps in establishing the proof of a given statement) by drawing auxiliary diagrams and then getting convinced of the truth/falsity of the proposed assertion by checking its validity, in many instances, after randomly dragging some elements of the figure (see Hohenwarter, Kovács, & Recio, 2017).

Moreover, DGS can help students making conjectures about a certain construction by allowing them to drag some of its free objects and observe the behaviour of those that depend on them. This dynamic visualization could enhance approaches listed in Polya (1962) such as the “pattern of two loci”—one of Polya’s four “patterns of thought.” An even simpler example of the potential relevance of dragging and observing could be the study of the relation between segments  $AP$  and  $BP$  while  $P$  moves in a circle with diameter  $AB$ .

A more sophisticated context of geometric reasoning involving DGS basic features arises when the user does backward conjecturing; that is, if the user makes, first, a certain construction and then states some property that he/she wants to hold true over the resulting figure. Obviously it does not hold true in general and the user has to look for necessary changes in the construction. Dynamic geometry programs facilitate the dragging of free objects and therefore the possibility of conjecturing how to restrict the position of these objects for the desired property to emerge. For example, consider a segment  $AB$  and a free point  $P$ . The user might want to find where to place  $P$  so that  $AP$ ,  $BP$  are perpendicular. By measuring the angle  $\angle APB$  and tracing the movements of  $P$  so the angle keeps close to  $90^\circ$ , the user can conjecture that  $P$  should be placed in a circle with diameter  $AB$ .

All of these useful features—and their associated impact in the classroom—belong, in some sense, to the past. Now, with GeoGebra ART, we can go much further—both for conjecturing that some property actually happens in a given construction and for discovering how to modify the construction for a given property to hold. A first novelty is that GeoGebra can take the lead and formulate the conjectures by itself. Thus, the program can, by simply comparing some geometric objects in a particular instance of a construction by means of the *Relation* tool, suggest a property holding between these elements as a conjectural truth. For example, in the above example with  $P$  located in a circle of diameter  $AB$ , we can directly ask GeoGebra for some relation between segments  $AP$ ,  $BP$  produced by a concrete position of  $P$  in the circle, yielding, as output, that for this particular instance, and computing with the numerical coordinates of  $A$ ,  $B$ ,  $P$ , the two segments seem to be perpendicular.

A second improvement is that, in this context, GeoGebra can automatically verify the general validity of this conjecture. In the affirmative case, it uses computation

with symbolic coordinates to establish a mathematical proof; that is, a proof not based on visual verification, approximate numerical calculations, or probabilistic analysis. To avoid confusion, the proof is performed in the program background and not shown to the user since it could fill hundreds of pages of algebraic formulas. Eventually, a message highlighting the need to avoid some degenerate cases for the truth of the given statement appears on the screen.

If the answer to the conjecture is negative and the user insists on requiring that the established conjecture holds, he/she can ask GeoGebra via the *LocusEquation* command for the precise formulation of necessary changes to the statement hypotheses.

We situate this workflow in a technology-mediated paradigm where the machine behaves like a mentor in the learning process, helping students in the intermediate steps of developing their own explanations of the truth of some geometric facts and fostering a *creativity spiral* as new discoveries are made by restarting the workflow again and again.

In the next section, we present a short overview of maths apps and programs related to dynamic geometry and theorem proving. We also give a basic description of the mathematical algorithm and system requirements behind our GeoGebra ART. In section three, we explore the use of these tools in education to help students discover geometric facts both experimentally and by proving. Section four focuses on a particularly novel feature of ART—the automatic discovery of missing hypotheses for a given (false) statement to become true. We also emphasize how this feature can help students “create” interesting, new geometric statements. Finally, section five collects some arguments and conclusions on the possible effect of GeoGebra ART in the learning and teaching of geometry.

## 2 Currently Available Maths Apps: A Short Overview

No doubt, there is a growing interest in developing convenient maths applications for contemporary students who are very experienced in using new technology, including smartphones and tablets. One of the most vibrant places to experiment with what the new technology offers is the market of Android applications, in particular those that are freely available and at the disposal of a very large set of users. Here we refer to a recent survey by Corpuz (2017) that summarizes some of the most popular Android and iOS apps as of October 2017. Corpuz’s collection consists of flexible scientific calculators for special purposes such as scientific or financial inputs, equation solvers, graphing calculators, and computational knowledge engines. Some of these popular software tools support handwriting recognition and/or capturing and interpreting camera shots of handwritten/printed formulas. On the other hand,

... when it comes to mathematics, it isn’t just getting the final answer that’s important; if anything, correct step-by-step solutions are far more important when it comes to teaching and learning math. (Corpuz, 2017, slide 9)

In this context, let us remark that one of the examples in the survey, the *Mathway* app, is able to provide the user not only with the answer to a problem (mainly in Linear Algebra, Calculus, or Trigonometry, introduced by typing or scanning handwritten input via the smartphone camera) but also with each step of the solution.

*Photomath* is not listed in Corpuz's collection but was the most popular maths app on *Google Play* in March 2018. *Photomath* provides step-by-step solutions to a comparable range of problems to the *Mathway* app free of charge in 36 languages.

It is, however, incautious to conclude that these examples represent a real change in the educational paradigm, opening the door to a new approach based on *computer-mediated thinking*. In fact, these apps are just enlarged pocket calculators that closely follow school traditions. Some of the answers they provide, such as the detailed steps in solving a linear univariate equation, are a usual requirement in the school curriculum in many countries.

On the other hand, step-by-step solvers for other fields of the maths curriculum (such as geometry) are not yet so popular. There are, however, remarkable attempts in some other areas such as:

- *Edukera* ([www.app.edukera.com](http://www.app.edukera.com)) that teaches step-by-step proving in the fields of logic, sets, calculus, and analysis,
- *Euclidea* ([www.euclidea.xyz](http://www.euclidea.xyz)) that teaches Euclidean geometry constructions as puzzles (120 puzzles are provided), with some basic dynamic geometry features.

These software tools are, however, only useful for finding solutions to pre-programmed problems. Even though the input seems to be a wide-open set of formulas, they are limited to solving a concrete set of close-ended problems. Thus, what they offer for a substantial part of mathematical activity—namely, for *discovery*—is very limited. Of course for most learners, these wonderful pieces of software are still inexhaustible, and their mathematical activity indeed simulates a kind of endless discovery.

In the following, we focus on a radically different approach. In our proposal, discovery plays a key role and the questions being posed challenge not only the underlying software, but also the user—both human and machine collaborating in the learning of something new!

Let us quickly recall that automated deduction of known or not yet discovered geometric results has a wide literature going back to the appearance of the first computers (see Botana et al., 2015 for a detailed overview). In particular, in planar Euclidean geometry, the first successful attempts in the 1950s (see Gelernter, 1959) led to a line of work in Formal Logic and Artificial Intelligence. As mentioned above, another important approach was based on Algebraic Geometry methods and was started by Wu (1978) and his early followers, including Chou (1987), and—focusing on the Gröbner bases method—Kapur (1986), and Kutzler and Stifter (1986), among others.

This is precisely the method used in the current GeoGebra implementation and that we roughly describe as follows: first, geometric statements are internally translated in terms of algebraic equations. For example, assume that the translation of the

hypotheses  $H$  and the thesis  $T$  of a given statement  $\{H \Rightarrow T\}$  is a collection of polynomial equations  $H = \{h_1 = 0, \dots, h_r = 0\}$  (respectively  $T = \{f = 0\}$ ). Then, the geometric instances verifying the hypotheses (respectively the thesis) are identified with the solution set of the corresponding polynomial system. And, if a non-empty and large subset (technically speaking: if a Zariski-open subset) of the algebraic variety of solutions of  $H$  is contained in the solution set of  $T$ , the theorem is labelled as *generally true*. This key inclusion test is performed by using Elimination procedures with Gröbner basis and checking whether the result is zero or not (see Recio & Velez, 1999 for further technical details).

DGS became very popular and well known with the breakthrough of personal home computers. Beginning with the *Geometric Supposer* (Schwartz & Yerushalmy, 1983) in 1981, widely used tools such as *The Geometer's Sketchpad* (Jackiw, 1995), *Cabri Geometry* (Baulac, Bellemain, & Laborde, 1994), and *Cinderella* (Kortenkamp, 1999) were available commercially. Another breakthrough, *GeoGebra's* (Hohenwarter, 2002) free availability for millions of users, opened the road to considering dynamic geometry as a natural education tool in the classroom.

Combining automated deduction in geometry (ADG) and DGS was a somewhat newer topic, but was already present in the 1990s as more than a research concept in the first versions of several DGS (de Villiers, 1999). These pioneer approaches, however, mostly used numerical and statistical methods for verifying properties holding among elements of a geometric figure. Just a few years later, in the second half of the 1990s, software tools appeared that used pure symbolic methods to prove or visualize geometric facts (see Botana & Valcarce, 2002; Oldenburg, 2008; Ye, Chou, & Gao, 2011).

We emphasize that a further requirement was essential to making substantial improvements in harnessing the possibilities of DGS; namely, accessing symbolic computations reliably and quickly, and connecting them dynamically with geometric objects. This is now possible in GeoGebra thanks to Giac (Kovács & Parrisé, 2015). GeoGebra is currently able to perform symbolic computations that allow the user to manipulate (simplify, expand, etc.) algebraic expressions in a way as rigorous as any well known computer algebra system.<sup>1</sup>

Finally, the application of these symbolic computations tools to automatic reasoning in geometry in GeoGebra was initiated by Recio, Botana and Abánades in 2010, and continued by many others, including Kovács, Weitzhofer, Parrisé, and Sólyom-Gecse over the last years (for more precise technical information and further references see Botana et al., 2015; Kovács & Parrisé, 2015). The outcome of this work, *GeoGebra ART*, will be discussed in the following sections.

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Computer\\_algebra\\_system](https://en.wikipedia.org/wiki/Computer_algebra_system).

### 3 Discovery and Creativity

Corless (2004) describes *computer-mediated thinking* by citing Peter Jones' notion of "intelligent partnership" (Jones, 1996) between the student and the computer (see Martinovic, Muller, & Buteau, 2013 for a detailed description). Corless mentions several examples of significant computational results achieved during this fruitful partnership. They are not only non-trivial for undergraduate students, but also surprising for many researchers, too. In fact, as Corless recalls by quoting the Portuguese Jewish philosopher, Abarbanel, mathematical discovery *should be a surprising activity*:

"I have absolutely no interest in proving things I know are true." (Corless, 2004, p. 10)

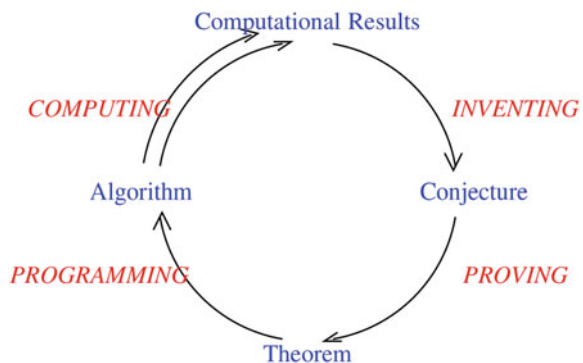
The relevance of student-computer cooperation in apprehending mathematical ideas through discovery is also emphasized in Buchberger's graphic *creativity spiral* describing the learner's workflow (Buchberger & The Theorema Working Group, 1998, see Fig. 1).

According to this concept, a continuous workflow can be identified starting with *computational results*. These results lead by intuition (or nowadays we could say by "big data" mining) to the invention of new *conjectures* that, in turn, yield to the formulation and eventually the proof of new *theorems*. From there, as Buchberger emphasizes, these results could lead to new *algorithms* and, by programming these algorithms, to new *computational results* about some mathematical fact. The spiral then continues with further inventions and conjectures (see also Kovács, 2018).

Of course, this process describes not only the learner's attitude to knowing mathematics better, but the researcher's position as well. In fact, both could be considered as quite similar, if we recall Halmos' idea (Halmos, 1982, p. vii) that "the only way to learn mathematics is to do mathematics."

Let us emphasize that both Corless' notion and Buchberger's spiral assume that the discovery process requires a computer. Thus, in the following, we will apply these basic principles of discovering geometric facts to the mediation of DGS. Also, we suppose—following the Jaworski's concept on the *teaching triad*: management of

**Fig. 1** Buchberger's concept, the creativity spiral



learning, sensitivity to students, and mathematical challenge (Jaworski, 1994)—that the related learning process will involve the presence of a human *mentor* as well. The mentor serves as a “personal coach and influencer” who attempts to affect the way a student behaves during a geometric task by suggesting concrete activities for finding the solution. On the other hand, as with sports coaches, we think it is useful to minimize the role of the teacher in the discovery process. The role of teachers in this new context is a delicate issue that needs precise research. One example of such research (described in Martinovic & Manizade, 2014) shows future teachers exploring activities designed to develop both technological and geometric skills.

We summarize our proposal for a novel approach to geometry learning through the use of the newly implemented ART in GeoGebra (see Kovács, Recio, & Vélez, 2017), as follows:


1. (a) The teacher provides the students with a kind of demo or tutorial on the use of DGS automated reasoning tools.  
(b) The teacher poses a problem. The nature of the problem is an open-ended question (in the sense of Foster, 2013) like “find all points  $P$  in the plane that have a certain property” (an “implicit locus problem”). More generally, we are thinking of questions for which the student has not been taught to follow a predetermined path for finding a solution. Of course, this is a very subjective situation. Using computers to “discover” shows that there is indeed a predetermined path to success, but not for the human user!
2. Some *computations* are performed with the DGS, based on a construction made by the student. In many cases this results in using the software tool for either random experiments or for experiments planned by the teacher. As an example, we will focus on computing a particular implicit locus (say, a circle).
3. A *conjecture* describing the characteristics of the output curve (e.g., a circle going through the three vertices of a given triangle) is made by the student.
4. The conjecture is checked numerically and symbolically by the DGS. We accept this result without further verification as a basic step. We now have a *theorem*. If applicable, the proof can be worked out by paper and pencil as well.
5. The next activity suggested by Buchberger’s spiral—the “programming” step—can be interpreted here as using the obtained result as a “piece” towards achieving more involved statements. These new pieces can then be assembled into new *algorithms*.

In this way, the obtained theorems and algorithms could be considered as a mere step towards the design and execution of “further experiments” by the student, whether controlled by the teacher or not, involving new *computations* with DGS (Step 1, second round). The process of Buchberger’s spiral then continues with Step 2.

## 4 An Example: Implicit Loci in GeoGebra

Using the steps above we now give some examples of how GeoGebra ART supports the discovery of an implicit locus. Let us consider the following simple implicit locus problem: *Given a line segment, describe the position of all points that are equidistant from the segment's endpoints.*

The solution to this basic introductory question is well known, both theoretically and through exploration with a DGS. In fact, this “direct” locus problem can be considered as an elementary instance of Voronoi diagrams, well known for their manifold real-life applications (Lindenbauer & Reichenberger, 2015). Yet, the teacher may propose this problem in an open-ended form (Step 0 (b)).

For the student, Step 1 may be to draw a triangle in GeoGebra by using the *Polygon* tool . By default, triangle  $ABC$  with sides  $a$ ,  $b$  and  $c$  is created (here and in the rest of the paper we assume that sides  $a$ ,  $b$  and  $c$  are opposite vertices  $A$ ,  $B$  and  $C$  respectively). Now the student can consider points  $A$  and  $B$  as fixed during the observation, and  $C$  as arbitrary and freely draggable. In the *Algebra View* it can be observed how side lengths  $a$  and  $b$  change when point  $C$  is dragged.

After collecting the results for a sufficiently high number of positions of  $C$ , a conjecture about the locus of  $C$  can be made. For some students this will be easy, but for others, probably not. To help the latter, additional colouring of the trace of  $C$  when  $a$  and  $b$  are close to each other could be enabled (Fig. 2, see Losada, 2014 for more details). This visual help may be unnecessary in this simple example, but for more difficult questions it could be quite relevant (see, e.g., Losada, Recio, & Valcarce, 2011).

Alternatively, GeoGebra can numerically compute the solution for the particular case when  $A$  and  $B$  are fixed. The student just needs to enter the command `LocusEquation(a==b,C)`<sup>2</sup> to obtain the algebraic equation of the resulting geometric curve; that is, the bisector line of segment  $AB$ , with the equation  $-8800x + 250y = -8607$ . Figure 3 shows the result.

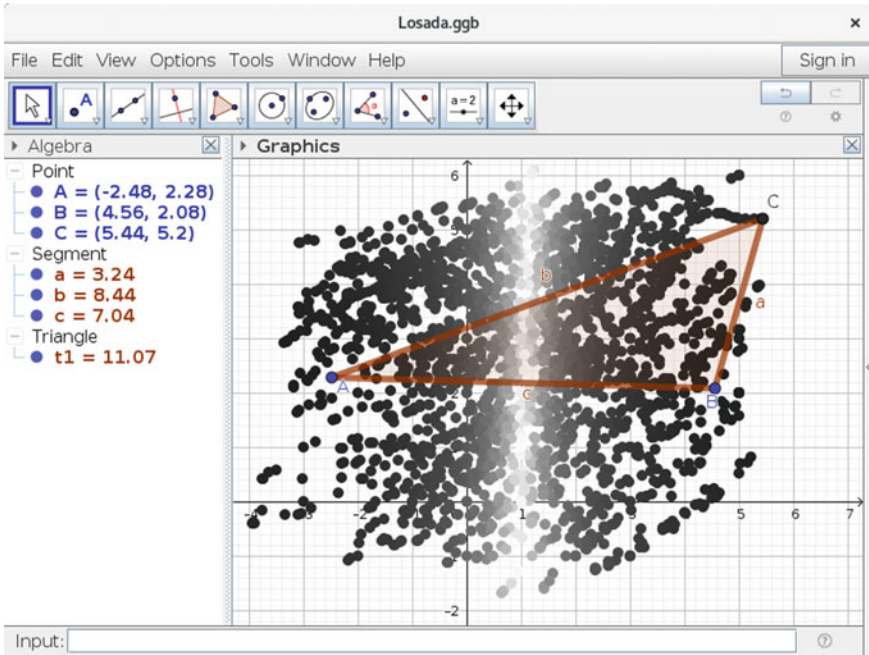
Of course, learners need to have some minimal knowledge of analytic geometry to be able to recognize that the obtained curve is indeed a line. This is, however, not strictly required. The conjecture could instead be formulated by simply observing the visual output of the locus in the *Graphics View*.

To confirm this, the student will also want to drag the points  $A$  or  $B$ . This could generalize the obtained result by creating several other setups of the input points. Figure 4 shows a few particular cases where  $A$  was moved from the original position to the position  $(-2, 3)$ . Of course, tracing should not be enabled in this step. It was done here to demonstrate the number of video frames shown during a short dragging interval.

---



<sup>2</sup>Notice that, according to the syntax of GeoGebra, the equation sign must be entered *twice*; this information is available in the GeoGebra Help or in Kovács (2017), but should also be remarked by the teacher in Step 0 (a).






**Fig. 2** Colouring the plane to find the locus. Here tracing is enabled for point C and *Dynamic Colors* are set for each RGB component by using the formula  $\frac{1}{|a-b|+1}$

In the final position, we obtained a linear equation again (here  $-328x + 46y = -303$ ). Hopefully, most students have the correct conjecture at this point; namely, that the locus is the perpendicular bisector of AB.

To continue with Buchberger’s schema, the student then goes on to Step 3—proving the conjecture numerically and symbolically. This substantial step requires a somewhat different construction. After disabling or deleting the obtained implicit curve  $d$  in the construction, the student should create the perpendicular bisector  $f$  of AB by using the *Perpendicular Bisector* tool . Then, by attaching an arbitrary point D on  $f$ , and—by using the *Segment* tool —creating segments  $g = AD$  and  $h = BD$ , point D will be freely draggable on  $f$  only and the *Algebra View* will show up-to-date information on the lengths of  $g$  and  $h$ .

Finally, using the *Relation* tool  with respect to  $g$  and  $h$ , the student tests (first, numerically) the validity of conjecture about the equidistance of all the points in the perpendicular bisector to the extremes of the segment. The *Relation* tool asks for any potential relation between these two segments.

Things are actually more complicated than simply concluding that  $g = h$  in all cases. Figure 5 shows an apparent counterexample concerning the lengths of these two segments after setting *Options/Rounding* to *15 Decimal Places*. Luckily, when comparing them by with the *Relation* tool, GeoGebra knows that a difference in

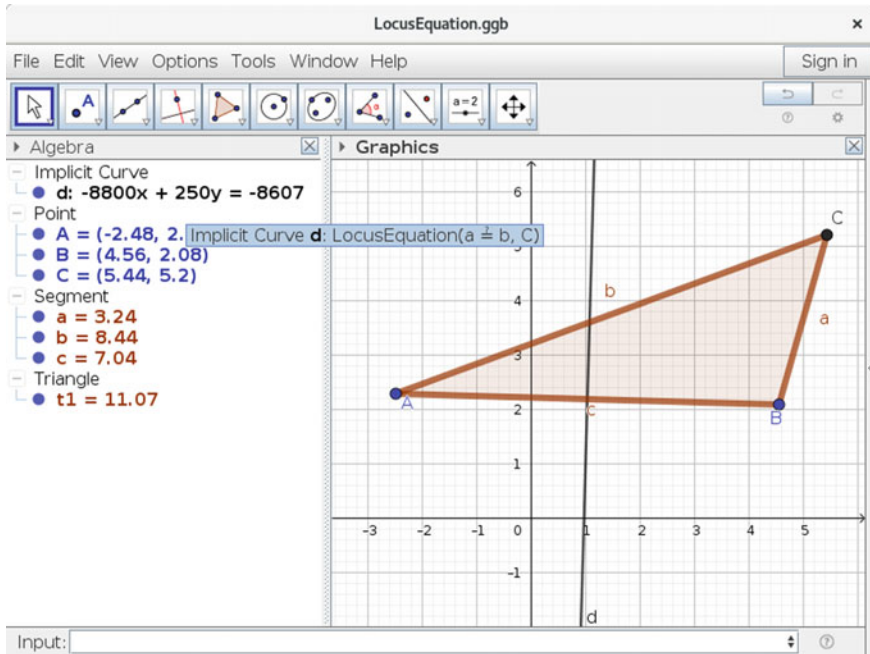


Fig. 3 Computing the locus equation in a particular case

precisely the last digit may be just a numerical error and assumes—just for the output of *Relation*—that the numerical comparison resulted in equality (Fig. 6).

Finally, overcoming all these potential numerical inaccuracies, by clicking the button “More...”, a symbolic check (that is, a mathematically rigorous proof) rigorous proof will be performed. ‘Symbolic’ here means a lot. Indeed, the input data are not anymore points with numerical coordinates, but with coordinates expressed by means of variables. The construction steps do not yield equations with numerical coefficients, but parametric equations depending on the parameters describing the coordinates of the free points, etc. Then, using sophisticated computer algebra algorithms, a complete proof consisting of up to a few millions of algebraic steps is done in the background. Only the final result “always true” is shown to the user (Fig. 7; see Kovács, 2015 for details).

Now the student has reached a point where there is more than enough evidence, based on the DGS automated reasoning tools, that the searched locus is the perpendicular bisector. The teacher (or the student) may then want to find a non-automated proof, and also a chain of reasons *why* the theorem holds. In this paper we do not focus on how this could be done in the best way.

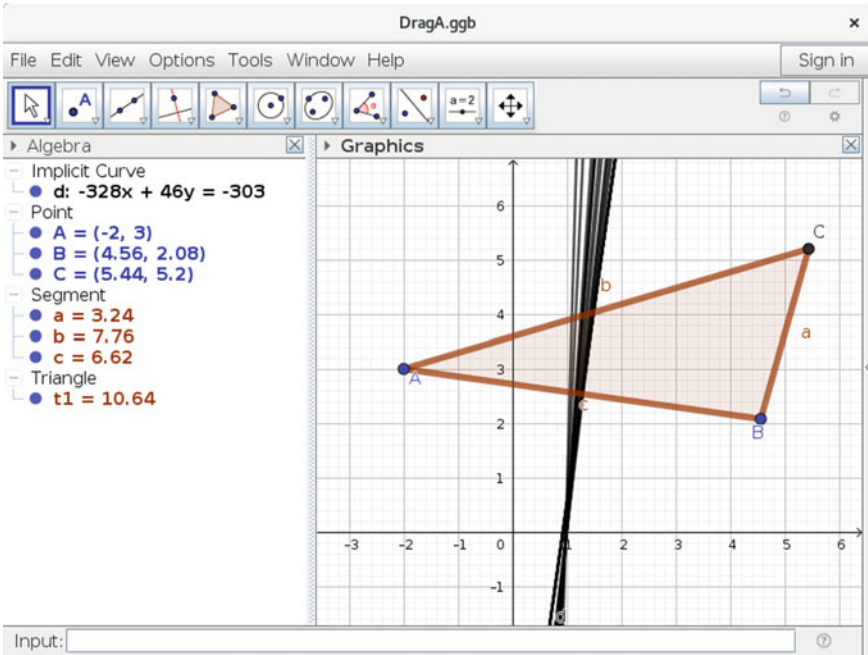


Fig. 4 Computing the locus equation for more than one case

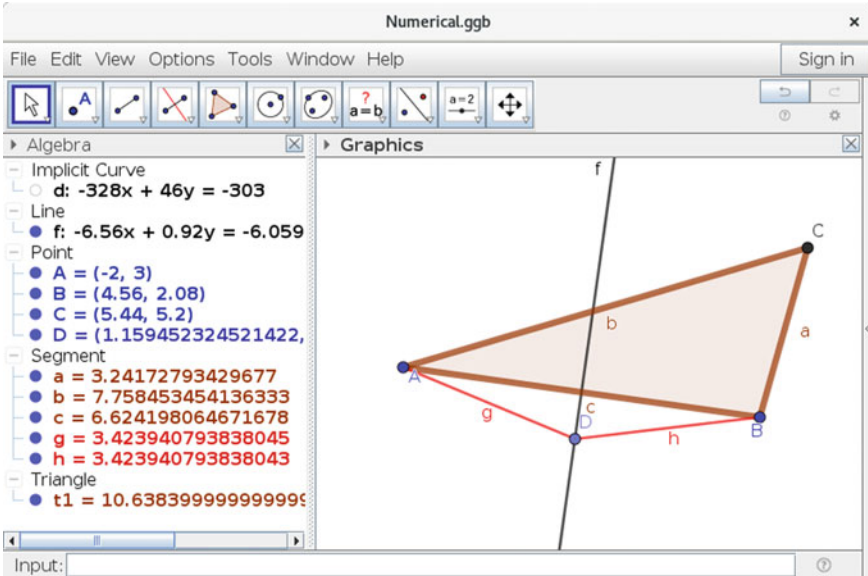
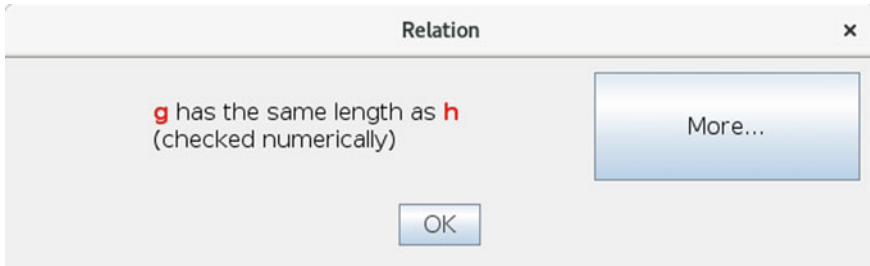
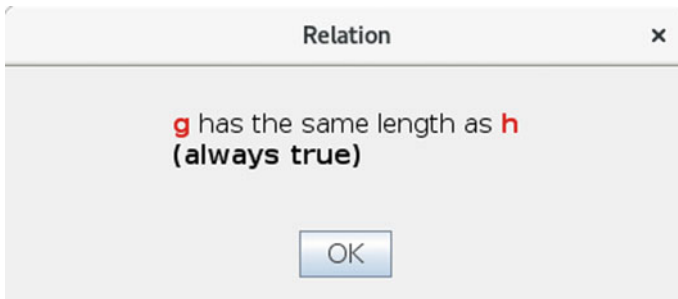


Fig. 5 Rounding errors may lead to confusing results



**Fig. 6** The *Relation* tool assumes that a minor numerical error may still not be contradictory to the conjecture



**Fig. 7** Symbolic check in the *Relation* tool

#### 4.1 A Second Round

Instead, we suggest going for a second round in the creativity spiral. Step 4 is about “programming”—here it could mean constructing a new figure or observing something different but related to the first round. There are several ideas to think about, but we will focus on the following: *Can we change the proposed statement by using a different formula than  $a == b$ ?* This idea involves just a minimal modification of the context, but results in a surprising change of the output. For example, let us consider the formula  $a == 2b$ .

The students can start again with Step 1 by doing various computations, including entering `LocusEquation(a==2b,C)`. Figure 8 shows the result.

It may be not obvious to the student what type of curve is shown as the output. Here the equation  $x^2 + 8x + y^2 - 4y = -4$  is shown, which is equivalent to the formula  $(x + 4)^2 + (y - 2)^2 = 4^2$ —clearly a circle. The student should, however, do several other attempts by dragging points *A* and/or *B* to verify other particular cases (Fig. 9). In the final position  $A = (3, 4)$ ,  $B = (4, 2)$ , the implicit curve  $3x^2 - 16x + 3y^2 - 28y = -80$  is obtained, which is, again, a circle, with a more difficult equivalent form. The intermediate curves also seem to be circles; that is, we may have a strong conjecture that the equation is a circle *in general*.

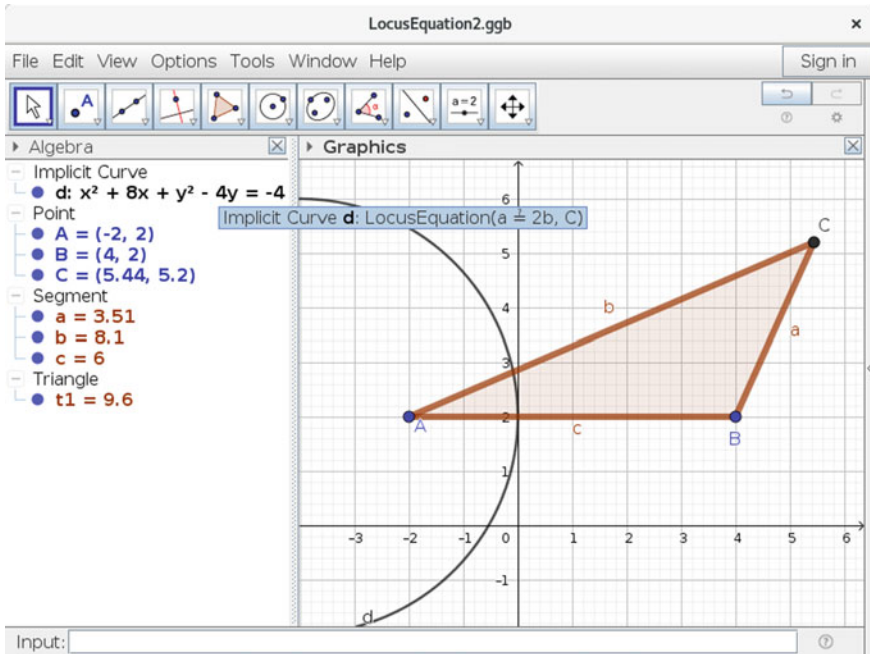


Fig. 8 Computing the locus equation in a particular case (second round)

We leave the reader to find the equivalent formula for the final circle in Fig. 9. It should help to discover the exact position of the conjectured circle. In addition, Fig. 10 discloses how the conjectured circle can be constructed by using a compass and a straightedge ruler. For some GeoGebra implementation issues, these tools are required to enable symbolic checks in the *Relation* tool.

Here we observe that the obtained circle is a *circle of Apollonius* (of Perga), with foci *A* and *B* and ratio 2.

### 4.2 Other Rounds

Clearly, other interesting questions can arise in this context, with surprising answers and involving unusual geometric facts. Here we only mention one simple possible modification of the previous formula; namely, considering `LocusEquation(a=2b+1, C)`, as shown in Fig. 11.

The set of obtained curves are of various geometry. In some cases, two closed curves are shown as the locus, but in other cases, just a subset of points in the locus output fulfills the formula requirements. For example, in Fig. 11 only the external curve provides points such that  $a = 2b + 1$ , while the internal one fulfills the

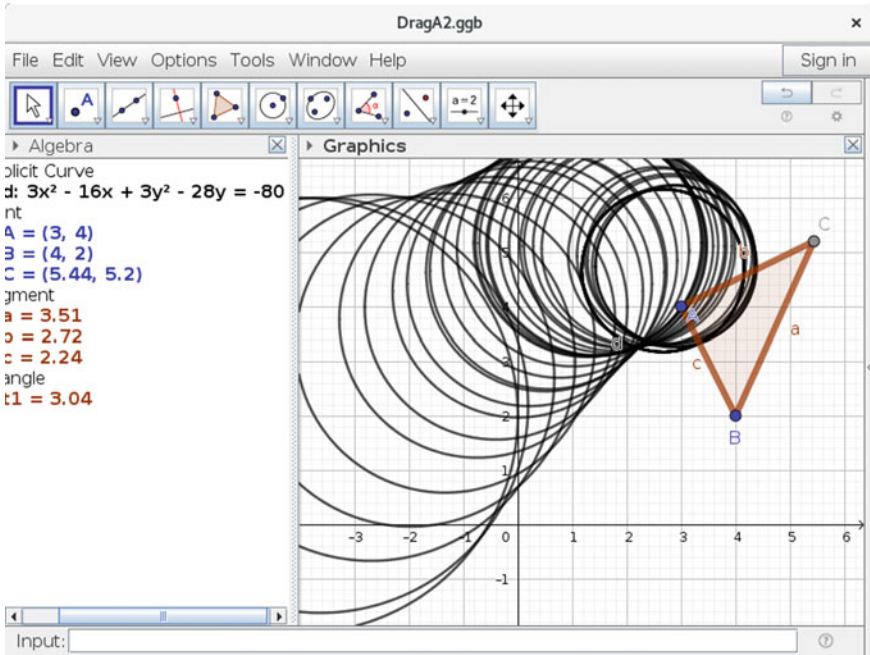


Fig. 9 Computing the locus equation in some particular cases (second round)

condition  $a = 2b - 1$ . The reason behind this unexpected effect has deeper roots in algebraic geometry and cannot be discussed in the classroom or in this chapter. Instead, we would like to emphasize that seemingly simple questions can actually lead to involved issues. In our opinion, however, this should not prevent us from *asking freely*—even more so, if our students ask questions on their own and get curious to discover the answer!

Finally, let us mention here the proposal of Krause (1975) that introducing a minimal change in the geometric assumptions of a given context results in a surprising shift towards a substantially new theory. We think the best educational setting happens when students are surprised by the answers they get and inspired to spend more time on discovering the richness of mathematics.

In this section we have discussed just one particular topic in planar geometry. However, many other topics can be raised in the classroom by using the “implicit locus approach”. Some typical curriculum topics include:

1. (The converse of) *Thales’ circle theorem* (see Artigue, 2012; Kovács & Schiffler, 2017): Given a segment  $AB$ , what is the locus of points  $C$  such that  $AC \perp BC$ ? (A generalization of this approach yields the inscribed angle theorem.)
2. A variation of the *triangle inequality*: Given a triangle  $ABC$  with side lengths  $a$ ,  $b$  and  $c$ , where the points  $A$  and  $B$  are fixed, what is the locus of points  $C$  such that  $a + b = c$ ?

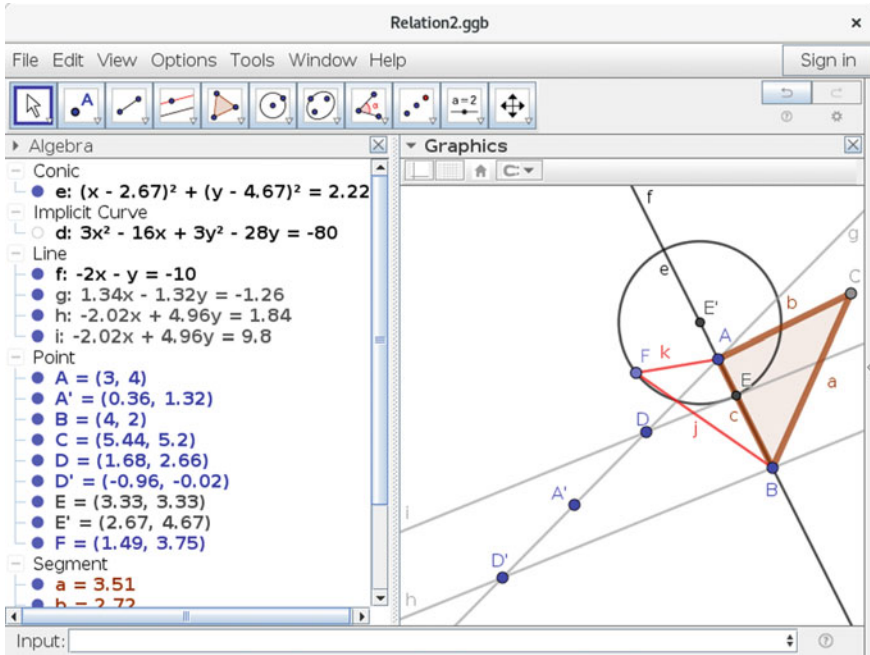


Fig. 10 Constructing the solution by using steps defined only by a compass and a ruler

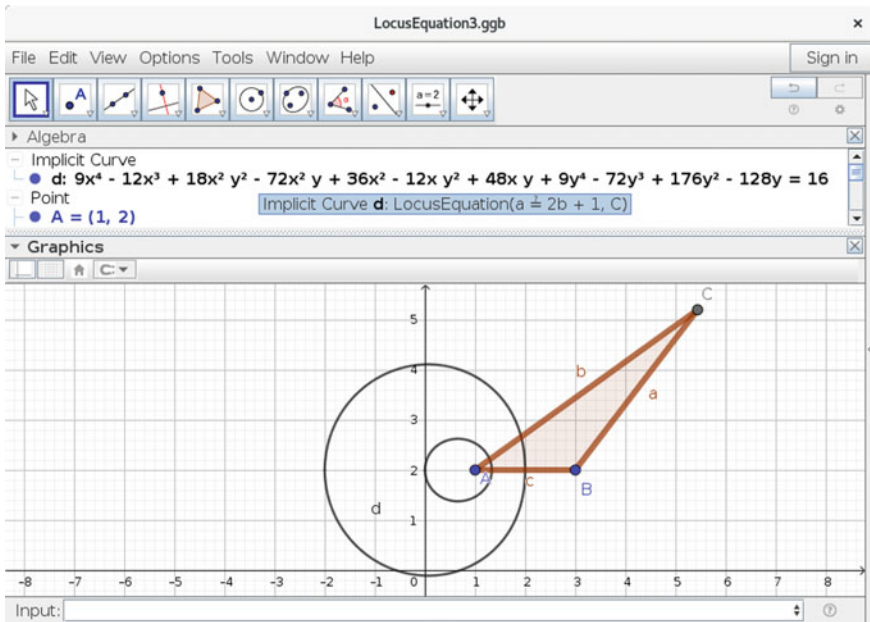
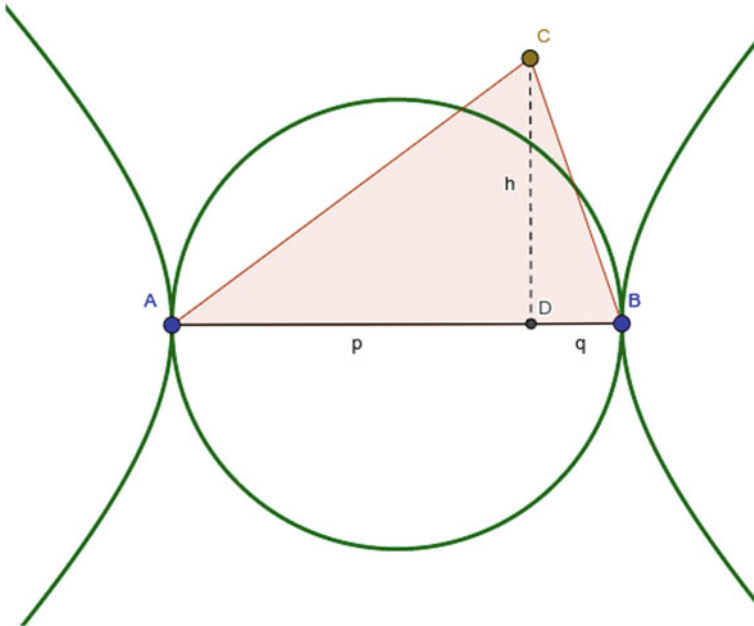


Fig. 11 Sometimes the locus equation can be a “yet unknown”, “strange” curve



**Fig. 12** GeoGebra ART detects an unusual locus when analyzing a traditional statement

3. (The converse of) *Pythagoras' theorem*: Given a triangle  $ABC$  with side lengths  $a$ ,  $b$  and  $c$ , where the points  $A$  and  $B$  are fixed, what is the locus of points  $C$  such that  $a^2 + b^2 = c^2$ ?
4. (The converse of) the *right triangle altitude theorem* (also known as *geometric mean theorem*): Given a triangle  $ABC$  with altitude  $h$  (with respect to  $C$ ) and two line segments  $p$  and  $q$  that it creates on side  $c$ , what is the locus of points  $C$  such that  $h = \sqrt{pq}$  (or, equivalently,  $h^2 = pq$ )? (See Fig. 12).

Let us highlight here that the obtained locus in the last example is not just a circle—according to the right triangle being assumed in the traditional theorem—but also a hyperbola (Abánades, Botana, Kovács, Recio, & Sólyom-Gecse, 2016a). This provides an immediate but not well known generalization of the right altitude theorem.

On the other hand, implicit loci are just a part of the new possibilities in GeoGebra ART. Other commands like direct *Prove*, *ProveDetails* or *Envelope* computation, are also available.

We refer to Botana (2016), Hašek (2017), Kovács and Vajda (2017), Kovács (2017, 2018), Abánades et al. (2016a, b), Kovács et al. (2017) for further details and examples of these recent methods in GeoGebra and to (Kovács et al., 2017) for full documentation.



## 5 Conclusion

It was stated 30 years ago in the futurist ICMI Study “School Mathematics in the 1990s” (Howson & Wilson, 1986) that

... even if the students will not have to deal with computers till they leave school, it will be necessary to rethink the curriculum, because of the changes in interests that computer have brought.

It was also long ago that an inspiring paper (Davis, 1995) by Philip Davis included a section specifically on the power of computer-based proofs to “transfigure” geometry statements and therefore to affect the potential role in mathematics education of software programs dealing with automatic theorem proving.

Another seminal paper in this context is Gila Hanna’s “Challenges to the Importance of Proof” (Hanna, 1995). Hanna considered the pedagogical and epistemological impact of different ways of using the computer to prove mathematical facts—for example, verifying a finite number of cases that remain to be checked for completing the proof of a statement, the so-called zero-knowledge proofs, and the case of proofs with a high probability of being correct. She also raised another challenge: the increasing habit of using the computer to establish the truth of a statement (visually or numerically) through experiments rather than formal proof.

Most of these messages about the relation between computers and proofs were essentially warning signs about issues that could take place in future worlds. Yet we think they are still quite present today. In fact, we see three basic concerns regarding dynamic geometry and mathematical reasoning. The first relates to the negative influence of the versatile visualization features of dynamic geometry programs. As Lin, Yang, Lee, Tabach, and Stylianides emphasize in (2012),

[The] increased availability in school mathematics instruction of ... dynamic geometry systems ... raised the concern that such programmes would make the boundaries between conjecturing and proving even less clear for students.

[They] allow students to check easily and quickly a very large number of cases, thus helping students ‘see’ mathematical properties more easily and potentially ‘killing’ any need for students to engage in actual proving.

Indeed, what is involved in the easy checking of a large number of cases is the “dragging” feature of DGS and, therefore, the above worries apply very particularly to all DGS.

The second concern relates only to the few DGS that currently provide automated reasoning tools, with at least the ability to confirm/deny the mathematical (not probabilistic) truth of a geometric statement. In the case of GeoGebra, as mentioned above, these tools are enlarged with other features for automatic discovery and beyond.

In fact, DGS with ART features can be considered “geometry calculators” and therefore as reflecting the already well known concerns about arithmetic or scientific calculators in mathematics education:

Can students be intellectually attracted to compute  $23456769 \times 98765432$ , once they know there is an algorithm that yields the correct answer 2316717923609208 and that it has been

implemented in their personal, say, tablet or phone? Likewise, will they be interested in finding whether the three heights of a triangle meet always at one point, if their pocket phone is able to guarantee, with a mathematical algorithm, that they certainly do so? (Hohenwarter et al., 2017)

Furthermore, we could ask: will they be interested in developing a local “deductive theory” (de Villiers, 1990) when insight into mathematical evidence does not seem to require this ability anymore, but, instead, a different kind of “cooperative reasoning” with the machine as described above?

The answer is unclear to us. It is possible the two contexts (arithmetic, geometry calculators) are not actually that parallel. In fact, the elementary geometry curriculum has always been deeply affected by the choice of “geometry tools” such as the ruler or the compass. Perhaps it is now the turn of a different geometry based on the existence and possibilities of Dynamic Geometry ART. Perhaps we should not just keep using new technology for old problems (old problems that are intimately “old tools”-driven). And perhaps we should consider the new technology as something other than a source of “concerns” (as outlined in Lin et al., 2012).

A third concern, going beyond the educational world, and considering GeoGebra ART just as a tool for the professional mathematician, has to do with one of Hanna’s “challenges to the importance of proof.” Although very relevant and nowadays increasingly present in professional math activity, the methods referred in Hanna (1995) for deciding the truth of a statement (using the computer to verify a large set of instances; zero-knowledge proofs; probabilistic proofs) have little to do with our approach, except for one collateral coincidence. In fact, as we have already emphasized, GeoGebra ART algorithms yield an exact (not numerically approximate, probabilistic, visual, or experimental) proof that would be accepted by the mathematical community as *if it were made by hand by a human*. This highlighted phrase is, perhaps, the crucial point. As Hanna poses in the above article:

Should mathematicians accept proofs that can not be verified by others?

Well, no, in our opinion. But, as described above, the main role we are proposing for the educational use of GeoGebra ART is not ‘acceptance,’ but “suggesting” ideas and ways for producing a formal, human readable, proof. The problem, of course, does not end with this declaration. We could also ask: what does “verified by others (humans)” mean today? Is using a calculator to verify a numerical computation such as  $23456769 \times 98765432$  forbidden in this context? Is “the verification” more acceptable only if the algorithm for this multiplication is done by hand on a piece of paper? How do we verify the correction of the algorithm and its implementation (by hand, by the human)? And, finally, if humans are allowed to use calculators to verify computations, should we restrict this to numerical calculators (and not to those performing algebraic computations)? Should the algorithmic expansion of  $(x + y)^{100}$  performed by GeoGebra on a mobile phone app be considered less “human verifiable” than the same computation done by a human on a blackboard?

We think there is a need to address urgently the extended impact of the merging, in one single tool, of the above three challenges related to DGS and Automatic Reasoning: (1) the potential, discouraging influence of powerful visualization features, (2)

the availability of an exact “geometry calculator”, and (3) the precise meaning today of “human readable” or “human verifiable” in a society where computers, laptops, and mobile phones are everywhere from kindergarten up.

Yet, the very recent survey (Sinclair et al., 2016) by Sinclair, Bartolini Bussi, de Villiers, Jones, Kortenkamp, Leung and Owens does not refer at all to automated reasoning tools. We think it is an urgent issue to address in future surveys of this kind, given the large expansion of GeoGebra in the classrooms worldwide—over 100 million users of its apps and website in 2017—and the fact that ART features have recently been included in it (see Kovács, 2015). This quantitative fact, indeed, has made a qualitative difference: as with pocket calculators, people will probably use ART for checking geometric facts with or without the consensus of the pedagogical community on its role.

We do not feel competent at this point to propose a concrete route towards geometry teaching and learning in this new context. We need feedback from the educational community (researchers, teachers, students) concerning what it means to *teach* and to *learn* and what *elementary geometry* means in this extended framework.

We simply feel GeoGebra’s automated reasoning capabilities can help our students to do mathematics better or faster, just as we think it is beneficial to have an electronic calculator to compute the square root of a number much faster than using the traditional, mechanical method by hand (which, not surprisingly, is no longer part of most curricula).

On the other hand, in the previous sections we have shown that even a simple question can yield difficult or surprising issues. It is in addressing these issues that mathematical creativity and reasoning can be fully developed but in a non-traditional way: human and machine equally ready to explore the geometric context. In this way we could argue that the intention of ART is not just to do the same kind of mathematics better and faster, but to do “a better kind of mathematics.” Let us borrow Kaput’s visionary words, cited by Balacheff (1997): instead of *doing (old) things better* we should focus on *doing better things*.

**Acknowledgements** We thank the referees for many interesting suggestions and comments and, in particular, for pointing us to several relevant bibliographic references. Special thanks to Gila Hanna, Dragana Martinovic, Chris Sangwin, Arleen Schenke, and David A. Reid for their direct help in improving the text of this chapter.

The second and third authors have been partially funded by Spanish and EDF Research Grant MTM2017-88796-P.

## References

- Abánades, M., Botana, F., Kovács, Z., Recio, T., & Sólyom-Gecse, C. (2016a). Towards the automatic discovery of theorems in GeoGebra. In G. M. Greuel, T. Koch, P. Paule, & A. Sommese (Eds.), *Mathematical Software—ICMS 2016. 5th International Conference*, Berlin, Germany, July 11–14, 2016, Proceedings. Volume 9725 of Lecture Notes in Computer Science (pp. 37–42). Cham: Springer International Publishing.

- Abánades, M., Botana, F., Kovács, Z., Recio, T., & Sólyom-Gecse, C. (2016b). Development of automatic reasoning tools in GeoGebra. *ACM Communications in Computer Algebra*, 50, 85–88.
- Artigue, M. (2012). What is inquiry-based mathematics education (IBME)? In M. Artigue & P. Baptist (Eds.), *Inquiry in mathematics education* (pp. 3–13). Fibonacci Project.
- Balacheff, N. (1997). ICME 8, TG19 followup report. Computer-Based Learning Environments: “CBILE”. <http://mathforum.org/mathed/seville/followup.html>.
- Baulac, Y., Bellemain, F., & Laborde, J. M. (1994). *Cabri geometry II*. Dallas: Texas Instruments.
- Botana, F., & Kovács, Z. (2016). New tools in GeoGebra offering novel opportunities to teach loci and envelopes. [arXiv:1605.09153](https://arxiv.org/abs/1605.09153).
- Botana, F., & Valcarce, J. L. (2002). A dynamic-symbolic interface for geometric theorem discovery. *Computers and Education*, 38, 21–35.
- Botana, F., Hohenwarter, M., Janičić, P., Kovács, Z., Petrović, I., Recio, T., et al. (2015). Automated theorem proving in GeoGebra: Current achievements. *Journal of Automated Reasoning*, 55, 39–59.
- Buchberger, B., & The Theorema Working Group. (1998). *Theorema: Theorem proving for the masses using Mathematica*. In *Invited Talk at the Worldwide Mathematica Conference*, Chicago, June 18–21, 1998.
- Chou, S. C. (1987). Mechanical geometry theorem proving. Springer Science+Business Media.
- Corless, R. M. (2004). Computer-mediated thinking. <http://www.apmaths.uwo.ca/~rcorless/frames/PAPERS/EDUC/CMTpaper.pdf>.
- Corpuz, J. (2017). Best math apps. <https://www.tomsguide.com/us/pictures-story/1300-best-math-apps.html>.
- Davis, P. (1995, March). The rise, fall, and possible transfiguration of triangle geometry: A mini-history. *The American Mathematical Monthly*, 102, 204–214.
- de Villiers, M. (1999). *Rethink proof with sketchpad*. Emeryville: Key Curriculum Press.
- de Villiers, M. (1990). The role and function of proof in mathematics. *Pythagoras*, 24, 17–24.
- Foster, C. (2013). Mathematical études: Embedding opportunities for developing procedural fluency within rich mathematical contexts. *International Journal of Mathematical Education in Science and Technology*, 55, 765–774.
- Gelernter, H. (1959). Realisation of a geometry-proving machine. In *Proceedings of the International Conference on Information Processing*, Paris, June 15–20, 1959 (pp. 273–282).
- Halmos, P. R. (1982). *A Hilbert space problem book* (2nd ed.). New York, Heidelberg, Berlin: Springer.
- Hanna, G. (1995). Challenges to the importance of proof. *For the Learning of Mathematics*, 15, 42–49.
- Hašek, R., Kovács, Z., & Zahradník, J. (2017). Contemporary interpretation of a historical locus problem with the use of computer algebra. In I. S. Kotsireas & E. Martínez-Moro (Eds.), *Applications of Computer Algebra: Kalamata*, Greece, July 20–23, 2015. Volume 198 of Springer Proceedings in Mathematics & Statistics. Springer.
- Hohenwarter, M. (2002). GeoGebra: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene. Master’s thesis, Paris Lodron University, Salzburg, Austria.
- Hohenwarter, M., Kovács, Z., & Recio, T. (2017). Deciding geometric properties symbolically in GeoGebra. In R&E-SOURCE (2017) Special Issue #6: 13th International Congress on Mathematical Education (ICME-13).
- Howson, G., & Wilson, B. (1986). *ICMI Study Series: School mathematics in the 1990s*. Kuwait: Cambridge University Press.
- Jackiw, N. R. (1995). *The Geometer’s Sketchpad*, v3.0. Berkeley, CA: Key Curriculum Press.
- Jaworski, B. (1994). *Investigating mathematics teaching: A constructivist enquiry*. Studies in Mathematics Education Series: 5. The Falmer Press.
- Jones, P. L. (1996). Handheld technology and mathematics: Towards the intelligent partnership (pp. 87–96). <http://ued.uniandes.edu.co/roless-calc.html>.
- Kapur, D. (1986). Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2, 399–408.

- Kortenkamp, U. (1999). Foundations of dynamic geometry. Ph.D. thesis, ETH Zürich.
- Kovács, Z. (2015). Computer based conjectures and proofs in teaching Euclidean geometry. Ph.D. thesis, Johannes Kepler University, Linz.
- Kovács, Z. (2017). Real-time animated dynamic geometry in the classrooms by using fast Gröbner basis computations. *Mathematics in Computer Science*, 11.
- Kovács, Z. (2018). Automated reasoning tools in GeoGebra: A new approach for experiments in planar geometry. *South Bohemia Mathematical Letters*, 25.
- Kovács, Z., & Parisse, B. (2015). Giac and GeoGebra—Improved Gröbner basis computations. In J. Gutierrez, J. Schicho, & M. Weimann (Eds.), *Computer algebra and polynomials*. Lecture Notes in Computer Science (pp. 126–138). Springer.
- Kovács, Z., & Schiffler, K. (2017). Unterstützung des Mathematikunterrichts mit automatischem Beweisen mit GeoGebra. In: PH forscht II, Linz, Austria.
- Kovács, Z., & Vajda, R. (2017). A note about Euler's inequality and automated reasoning with dynamic geometry. [arXiv:1708.02993v2](https://arxiv.org/abs/1708.02993v2).
- Kovács, Z., Recio, T., & Vélez, M. P. (2017). gg-art-doc (GeoGebra Automated Reasoning Tools. A tutorial). A GitHub project. <https://github.com/kovzol/gg-art-doc>.
- Kovács, Z., Recio, T., Richard, P. R., & Vélez, M. P. (2017). GeoGebra automated reasoning tools: A tutorial with examples. In G. Aldon & J. Trgalova (Eds.), *Proceedings of the 13th International Conference on Technology in Mathematics Teaching*. (pp. 400–404). <https://hal.archives-ouvertes.fr/hal-01632970>.
- Krause, E. F. (1975). *Taxicab geometry: An adventure in non-Euclidean geometry*. Addison-Wesley.
- Kutzler, B., & Stifter, S. (1986). On the application of Buchberger's algorithm to automated geometry theorem proving. *Journal of Symbolic Computation*, 2, 389–397.
- Lin, F. L., Yang, K. L., Lee, K. H., Tabach, M., & Stylianides, G. (2012). Principles of task design for conjecturing and proving. In G. Hanna & M. de Villiers (Eds.), *Proof and Proving in Mathematics Education. The 19th ICMI Study* (pp. 305–326). Springer.
- Lindenbauer, E., & Reichenberger, S. (2015). Voronoi-Diagramme. GeoGebra Materials. <https://www.geogebra.org/m/sAaFMcTA>.
- Losada, R. (2014). El color dinámico en GeoGebra. *La Gaceta de la Real Sociedad Matemática Española*, 17(3), 525–547.
- Losada, R., Recio, T., & Valcarce, J. L. (2011). Equal bisectors at a vertex of a triangle. In B. Murgante, O. Gervasi, A. Iglesias, D. Taniar, & B. Apduhan (Eds.), *Computational Science and Its Applications—ICCSA 2011*. Lecture Notes in Computer Science (Vol. 6785, pp. 328–341). Berlin, Heidelberg: Springer.
- Martinovic, D., & Manizade, A. G. (2014). Technology as a partner in the geometry classrooms. *The Electronic Journal of Mathematics and Technology*, 8, 69–87.
- Martinovic, D., Muller, E., & Buteau, C. (2013). Intelligent partnership with technology: Moving from a math school curriculum to an undergraduate program. *Comput. Schools*, 30, 76–101.
- Oldenburg, R. (2008). *FeliX - mit Algebra Geometrie machen (German)* (pp. 15–17). Computeralgebra Rundbrief: Sonderheft zum Jahr der Mathematik.
- Polya, G. (1962). *Mathematical discovery: On understanding, learning, and teaching problem solving*. London, UK: Wiley.
- Recio, T., & Vélez, M. P. (1999). Automatic discovery of theorems in elementary geometry. *Journal of Automated Reasoning*, 23, 63–82.
- Schwartz, J. L., & Yerushalmy, M. (1983). *The Geometric Supposer*. Pleasantville, NY: Sunburst Communications.
- Sinclair, N., Bartolini Bussi, M. G., de Villiers, M., Jones, K., Kortenkamp, U., Leung, A., et al. (2016). Recent research on geometry education: An ICME-13 survey team report. *ZDM Mathematics Education*, 48, 691–719.
- Wu, W. T. (1978). On the decision problem and the mechanization of theorem-proving in elementary geometry. *Scientia Sinica*, 21(2). Reprinted In W. W. Bledsoe, & D. W. Loveland (Eds.), *Automated Theorem-Proving: After 25 Years*. Providence, RI: AMS (1983).

Ye, Z., Chou, S. C., & Gao, X. S. (2011). An introduction to Java Geometry Expert. In *Automated Deduction in Geometry, 7th International Workshop, ADG 2008*, Shanghai, China, September 22–24, 2008. Revised Papers, Lecture Notes in Computer Science (Vol. 6301, pp. 189–195). Springer.