The portfolio prover in GeoGebra 5

Zoltán Kovács

Johannes Kepler University, Department of Mathematics Education
Altenberger Strasse 54, A-4040 Linz, Austria
zoltan@geogebra.org

Abstract. GeoGebra is open source mathematics education software being used in thousands of schools worldwide. Its forthcoming new version 5 will support automatic geometry theorem proving by using various methods which are already well known, but not widely used in an education software. GeoGebra's new embedded prover system chooses one of the available methods and translates the problem constructed by the end user as the input for the selected method, similarly to portfolio solvers. The applicable methods include Wu's method, the Buchberger-Kapur method, the Area method and Recio's exact check method, some of them as embedded algorithms, others as outsourced computations. Since GeoGebra maintains the development in an open-sourced way by collaborating with the OpenGeoProver, Singular and Giac projects as well, further enhancements can be expected by a larger community, including implementing other methods as well.

Keywords: GeoGebra, portfolio solver, computer algebra, computer aided mathematics education, automated theorem proving

1 Introduction

GeoGebra [1] is educational mathematics software, with millions of users worldwide. Its founder, Markus Hohenwarter broadened its software development into an open source project in 2005. GeoGebra has many features (including dynamic geometry, computer algebra, spreadsheets and function investigation), but it primarily focuses on facilitating student experiments in Euclidean geometry, and not on formal reasoning. Including automated deduction tools in GeoGebra could bring a whole new range of teaching and learning scenarios. Since automated theorem proving (ATP) in geometry has reached a rather mature stage, some ATP experts agreed on starting a project of incorporating and testing a number of different automated provers for geometry in GeoGebra. This collaboration was initiated by Tomás Recio in 2010.

It must be emphasized that a number of software systems have been existing for several years which focus on computer aided proving and dynamic geometry. The most well known include *GeoProof* [1] by Julien Narboux [2,3], *GDI-Discovery*

[1] http://home.gna.org/geoproof/

[4] by José Valcarce and Francisco Botana, the *Geometry Expert*[2] [5] by Shang-Ching Chou, Xiao-Shan Gao and Zheng Ye, and *GEOTHER* [6] by Dongming Wang. These (and some other) dynamic geometry systems (DGS) with ATP features can efficiently prove many complex geometry theorems, but these ATP features are not primarily designed for applications in education[3]. They are, in many aspects, still in the prototype phase, not yet well distributed, maintained or not fully operative.

On one hand, GeoGebra is the most widely used DGS in the mathematics education in the world. According to the webpage `http://99webtools.com/pagerank_tool.php` in February 2014 Google PageRank for GeoGebra was 7. This number was the same as for *Maple*, less than for *Mathematica* (8), and greater than for *Cabri*, *Cinderella*, *The Geometer's Sketchpad*, *DrGeo* (6), *WIRIS*, *Geometry Expressions* (5), *Geometry Expert*, *kig* and *Live Geometry* (4).[4] Also GeoGebraTube, available at `http://www.geogebratube.org`, GeoGebra's primary repository for freely available teaching materials, is a dynamically growing database with more than 90,000 of materials (as of June 2014).

On the other hand, a remarkable amount of work has already been contributed to publicly available algorithms and their implementations. Meanwhile also open-sourced ATP systems and test databases were introduced, including *OpenGeoProver* (OGP) [7] and *GeoThms* [8], started and maintained by Predrag Janičić, Pedro Quaresma and their colleagues. Collaborative work seemed to be a very important step in widening the availability of ATP algorithms to be used in education. We found that a primary problem is that scientific contributions to DGS and ATP are isolated, and in an open-sourced system the existing efforts could be summarized and continued.

In 2011 Narboux, Yves Bertot and his PhD student Tuan Minh Pham started to change GeoProof's user interface to GeoGebra [9], but they still used *Coq* [10] as the external formal prover. However, a part of the GeoGebra Team, lead by the author, advised by Hohenwarter, Recio and Botana, and supported by Janičić, started to work on a different approach at the beginning of 2012—their solution was to use both an embedded system in GeoGebra and also outsourced computations. This prototype has already been published at the EACA 2012 [11, 12] and CADGME 2012 [13] conferences.

In this paper a general overview is given about the technical details of the work of the GeoGebra developers related to this latter approach. In Section 2 arguments for the structural decisions are shown. In Section 3 the design of the portfolio prover is demonstrated. Section 4 shows some examples how the

---

[2] `http://www.mmrc.iss.ac.cn/gex/`

[3] One example of a system planned for teaching proofs interactively is Jacques Gressier's *Geometrix*, available at `http://geometrix.free.fr`.

[4] Similar statistics can be obtained from the Alexa web information company at `http://alexa.com`: the February 2014 results are: 9475. Mathematica, 78826. GeoGebra, 178487. Maple, 335343. WIRIS, 1288617. Cinderella, 2063304. Geometer's Sketchpad, 4425178. Geometry Expressions, 5295648. DrGeo, 13982880. Geometry Expert. kig and Live Geometry could not be reliably measured since they are hosted on KDE's and Codeplex's server (which are listed as 20989. and 2238. globally).

implemented system works. Section 5 discusses the possible ways of a future enhancement of the joint work.

2 Open source collaboration

One of the most advanced pieces of software being available for free download is the *Java* Geometry Expert (*JGEX* [14, 15]) developed by Chou, Gao and Ye, but built upon the work of several other experts of the modern Chinese computational mathematics. JGEX's prover system contains multiple methods to compute proofs of the input being constructed graphically or by a special programmatical language. The available methods are the Gröbner basis method (developed by Buchberger, Kapur [16] and others), Wu's method [17], the Area method [18] and the geometry deductive database (GDD) method [19].

JGEX gives the opportunity for the user to select from the existing methods since in some circumstances any of them can fail: each has its strengths and weaknesses. In GeoGebra's portfolio prover (PP) we followed the same way, but an automated selection will be used to minimize user interaction: we assume that the user is a secondary school student who knows nothing about ATP. That is, PP will work with default values which can be overridden by only expert users via command line arguments[5].

Unfortunately, the source code for JGEX is not available for free downloading, and also the GDD method (which probably has the most remarkable reputation on educational use) is a kind of black box. GeoGebra's long term success in the classroom is not only that it is freely available for end users, but it is also possible to be enhanced and bug-fixed—the open source development model is a requirement for that.

We also wanted to leave the opportunity for other DGS to use the implemented methods in other applications than GeoGebra as well. For such a warranty we agreed starting a joint work with University of Belgrade on the OGP system. In its first version OGP was capable only for computing geometry proofs by using Wu's method, but later we managed to get support from the Google Summer of Code project for a summer stipend of a French university student, Damien Desfontaines—he implemented the Area method in OGP in 2012 [20]. The OGP project was considered as a fruitful way for collaboration of other experts as well: the Mass point method and the Full angle method were started to work on (an introductory report on the latter was already published by Quaresma and Baeta in 2013 [21]).

Meanwhile, we also started to work on internal implementations of other methods. The first internally implemented method was a new approach suggested by Recio, to use exact coordinates and arbitrary integer arithmetics on testing geometry statements to obtain yes/no proofs for a set of elementary theorems. His algorithm [12] was implemented by the Austrian student Simon

[5] `http://wiki.geogebra.org/en/Release_Notes_GeoGebra_5.0#New_Command_Line_Arguments`

Weitzhofer in 2012 and published in his master thesis [22] in 2013. A more traditional approach, namely the Buchberger-Kapur method (based on Gröbner basis computation), was also implemented by the author, by using external computations for solving equation systems. This method was later extended to use the Recio-Vélez algorithm [23] to obtain degeneracy conditions, and the algorithm was later enhanced for better educational use.

The outsourced computations were achieved by using *Singular* [24] as an external web service, running on a dedicated virtual server. This technology was discussed by Botana and the author in the planning stage of GeoGebra's ATP capabilities. For this reason the applied method in GeoGebra is called "Botana's method" [11].

However in GeoGebra we already managed to make the prototype work, many schools and students started to move from the desktop application to a different technology: they preferred to use tablets and smartphones instead of desktop PCs and laptops. Unfortunately, the Java technology and the outsourced computations are not always applicable in the changed way of using computers in the education. That is why we had to find even new technologies to support the HTML5/JavaScript approach of application development, including offline HTML5 applications as well [25]. Fortunately, the GeoGebra Team managed to change the internal computer algebra system from *Reduce* [26] to *Giac* [27] which was a step forward to support faster Gröbner basis computations, also available in a web browser in offline mode on a tablet or a smartphone [28].

Now GeoGebra's PP is much more than a prototype. It is fully documented not only in its source code and the Developers' Howto[6] but on GeoGebra's Wiki pages[7], and a set of demonstrational examples is available on GeoGebraTube[8]. It is an extensible system in both GeoGebra and OGP.

We would like to highlight that the success of GeoGebra's PP is based definitely on its open-sourced roots. Without the existing knowledge in the implementations of several modern algorithms in the used systems—especially in Singular and Giac—GeoGebra would not have the chance to offer competitive ATP features for the mathematics education. These systems have several years of programmers' experience and millions of lines of program code which could not have been reimplemented from scratch within a reasonable time.

3 The design of PP

The design of PP in GeoGebra is shown in Fig. 1. On the top of the figure GeoGebra's user interface is demonstrated: the higher the action is drawn, the easier communication for the user is assumed. Also in former versions of GeoGebra the highest level action is to use the *Relation Tool* which purely decides if

[6] See `http://dev.geogebra.org/trac/wiki/TheoremProving` for more details. This documentation includes detailed description of the applied methods and the supported constructions and statements for them.

[7] For an example, see `http://wiki.geogebra.org/en/ProveDetails_Command`.

[8] See `http://www.geogebratube.org/student/b104296` for an example.

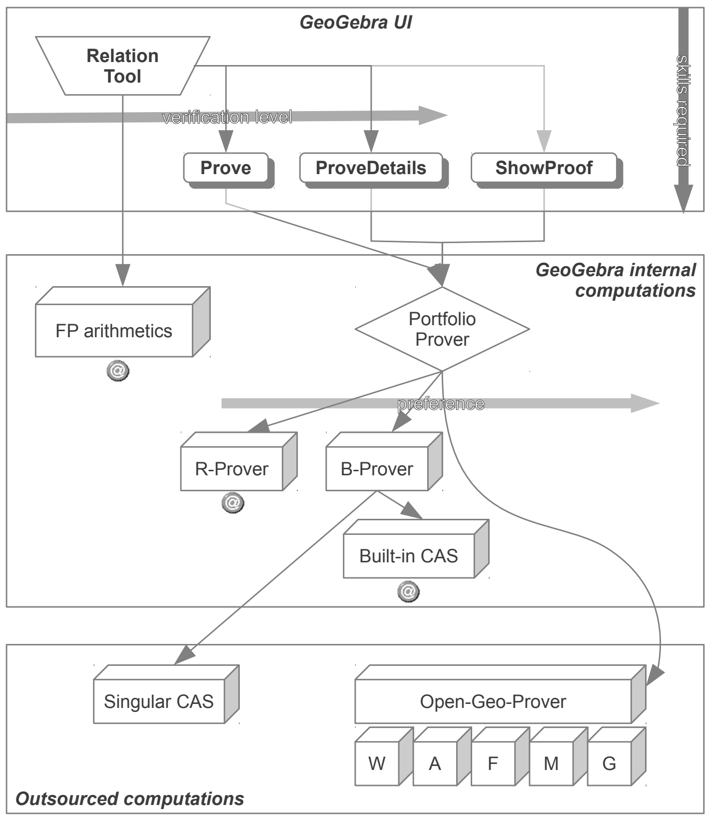

Fig. 1. The design of GeoGebra's portfolio prover

two geometrical objects have a relation like parallelism, perpendicularity, equal length etc. Prior to GeoGebra 5 Relation Tool was using floating point (FP) arithmetics to decide if a certain relation holds between the input objects. In some cases, however, FP returns the inaccurate or wrong answer, thus the output for Relation Tool is incorrect.

In such incorrect cases there is a need to increase the "verification level" of the investigation which means we collect more or more trustworthy information about the statement to decide. GeoGebra's new command **Prove** gives an ATP answer instead of FP arithmetics with the possible outputs "yes", "no" or "undefined" (the last kind of output means that no trustworthy answer was found). What is more, the new **ProveDetails** command can give more details on degeneracy conditions, that is it can refine the statement by adding a sufficient condition if needed. For the moment the **ShowProof** command is not yet implemented just planned: when a "readable enough" proof is found by the ATP subsystem, it could be shown to the student for ensuring complete certainty.

In the middle of Fig. 1 GeoGebra's internal computations are shown. PP optionally analyzes the look of the statement and tries to select the most fitting method to work with. If no analysis is used (that is, in AUTO mode), then a simple priority list of the methods is taken: at the moment for the **Prove** command this is

$$\text{Recio} < \text{Botana} < \text{Wu (OGP)} < \text{Area (OGP)},$$

and for the **ProveDetails** command[9]:

$$\text{Botana} < \text{Wu (OGP)}.$$

Recio's method (R-Prover) is a quick method for proving statements concerning points and lines in a triangle. Since it cannot be applied for constructions containing conics, and for more than 3 free points it may be too slow, GeoGebra will consider Botana's method (B-Prover) as a fallback in such cases. At the moment some construction types (including angle bisectors) are not yet implemented in the B-Prover, thus in such cases GeoGebra will go ahead by using OGP's Wu's method to obtain an ATP answer. If Wu's method fails, GeoGebra resends the statement to OGP by requesting the computation via the Area method—for example, to prove Ceva's theorem this will be the only working way at the moment (see Section 4.3): GeoGebra cannot properly describe yet the thesis statement for OGP's Wu's method to be suitable for processing, but there is an extra algorithm implemented only in OGP's Area method which accepts a wider set of statements.

R-Prover uses arbitrary integer arithmetics internally, but B-Prover requires computing solutions of a polynomial equation system for the **Prove** command and elimination for the **ProveDetails** command. (In general, B-Prover requires Gröbner basis computations.) Detailed investigation showed that it would be too time consuming to implement an internal algorithm in GeoGebra to efficiently

[9] Degeneracy conditions can be obtained only by two methods at the moment.

compute polynomial based calculations, thus we use the Singular computer algebra system (CAS) instead. After changing internal CAS of GeoGebra from Reduce to Giac we found that Giac computed Gröbner bases surprisingly efficiently, and its speed was comparable with Singular in many cases. Also Giac can easily be used in a web browser as well since it is written in C++ and by utilizing the *emscripten* [29] C++ to Javascript compiler (or Google's *NaCl* C++ to Native Client compiler) the Gröbner basis computations are still acceptably fast.[10]

In the bottom of Fig. 1 the externally used systems are shown. At the moment, these external systems cannot be used in HTML5 mode (including online and offline modes)—we used the symbol "@" to mark those subsystems in PP which are transparent for the technology change. Our future plans include compile both Singular and OGP to be technology transparent.

OGP currently supports Wu's method (W) and the Area method (A), and is subject to be enhanced by additional ATP methods including the Full angle method (F), the Mass point method (M) and the Gröbner basis method (G)—the last is definitely the same as the internal B-Prover in GeoGebra.

At the top of Fig. 1 the Relation Tool is shown as the easiest way to start GeoGebra's PP. For users having advanced skills, PP is also available via GeoGebra commands.

4 Examples

In this section four theorems are provided as shown in GeoGebra 5. Most of them can be introduced in many secondary schools and thus they are examples of possible classroom uses of GeoGebra's PP. The first three examples run in the desktop version, and the final example is shown in a web browser.

Despite the interesting part of the log messages are shown after these examples, they are not intended to be displayed neither for the students, nor the teachers. Here they are printed for the researcher's interest. Students and teachers should be informed via GeoGebra's user interface only—at the moment in the Algebra View by getting the output of some GeoGebra commands, and in a later GeoGebra version inside a dedicated popup window.

That is, the output of the **Prove** command is currently shown on the left of the GeoGebra window (by default) among the Boolean Value entries (see the top-left corner of Fig. 2, 3 and 4). The output of the **ProveDetails** command is shown among the List entries (see the bottom-left corner of Fig. 5).

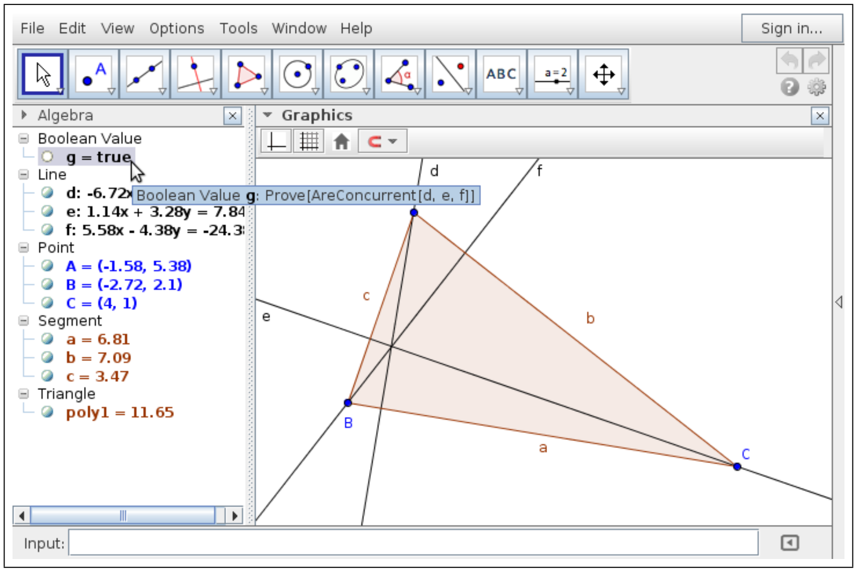

Fig. 2. Heights of a triangle are concurrent

4.1 Heights of a triangle are concurrent

Fig. 2 was created by using GeoGebra 4.9.257.0[11] by drawing a triangle with its heights d, e and f and then the command **Prove**[**AreConcurrent**[**d,e,f**]] was entered in the Input Bar (at the bottom). Here PP selects R-prover to compute $\binom{3+2}{2}$ tests for a trustworthy answer if the heights are always concurrent (not considering some degenerate cases). The computation took 8 milliseconds on a typical PC[12]:

```
12:32:26.218 DEBUG: geogebra.m.o$a.run[-1]: Using AUTO
12:32:26.218 DEBUG: geogebra.common.p.y.a[-1]: Using RECIOS_PROVER
12:32:26.219 DEBUG: geogebra.common.l.q.a.a[-1]: nr of tests: 10
12:32:26.224 DEBUG: geogebra.common.l.q.i.<init>[-1]: Benchmarking: 8 ms
12:32:26.224 DEBUG: geogebra.common.l.q.i.<init>[-1]: Statement is TRUE
```

4.2 Varignon's theorem

Fig. 3 shows an arbitrary quadrilateral with the midpoints of its sides (E, F, G and H). Now when considering the quadrilateral $EFGH$ we can find that it is a

[10] See http://test.geogebra.org/~kovzol/data/prove-provedetails-20140120/ for a recent report on benchmarking various theorems with the R-Prover based on computations with Singular and Giac, and compared with OGP's Wu's method.

[11] Beta versions of GeoGebra 5 are available at http://download.geogebra.org/installers/5.0.

[12] Java method names in the log are obfuscated to ensure faster results and a smaller software package.

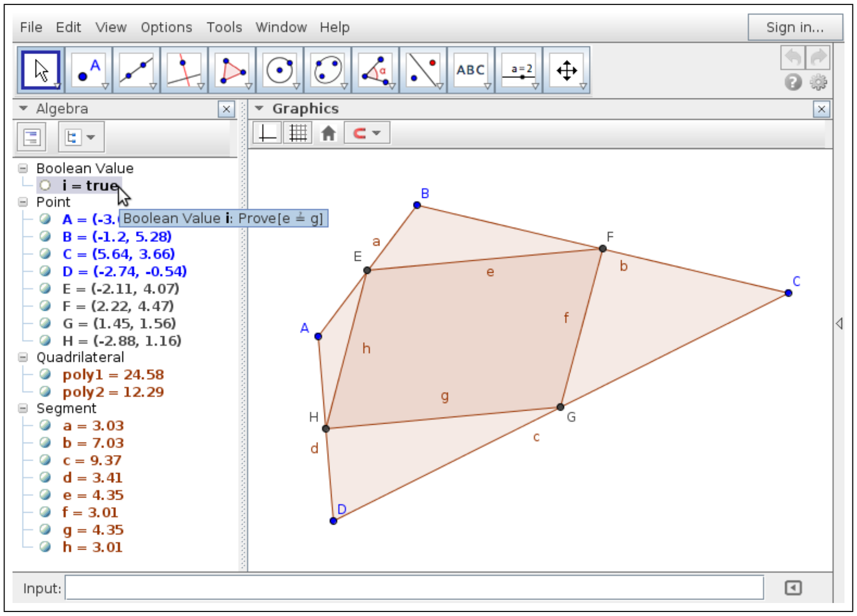

Fig. 3. Varignon's theorem

parallelogram. To get an ATP yes/no answer to verify this we need the command **Prove[e==g]**, for example. Since R-Prover cannot deal with Pythagorean distances, B-Prover is selected by PP. The computation took 47 milliseconds on a typical PC (which already contains the HTTP request to the external server and also its background computation—here the uninteresting parts of the log messages have been omitted and substituted by `[...]`):

```
12:56:23.790 DEBUG: geogebra.m.o$a.run[-1]: Using AUTO
12:56:23.790 DEBUG: geogebra.common.p.y.a[-1]: Using RECIOS_PROVER
12:56:23.791 DEBUG: geogebra.common.p.y.a[-1]: Using BOTANAS_PROVER
[...]
12:56:23.806 DEBUG: geogebra.common.p.y.b[-1]: Thesis reductio ad absurdum (denied statement):
12:56:23.806 DEBUG: geogebra.common.p.y.b[-1]: 9. -1+-1*v17*v16^2+-1*v17*v15^2+2*v17*v16*v14+
 -1*v17*v14^2+2*v17*v15*v13+-1*v17*v13^2+v17*v12^2+v17*v11^2+-2*v17*v12*v10+v17*v10^2+
 -2*v17*v11*v9+v17*v9^2
12:56:23.807 DEBUG: geogebra.common.l.q.n.a[-1]: ring r=(0,v1,v2,v3,v4,v5,v6,v7,v8),
 (v9,v10,v11,v12,v13,v14,v15,v17,v16),dp;ideal i=2*v9+-1*v3+-1*v1,2*v10+-1*v4+-1*v2,
 2*v11+-1*v5+-1*v3,2*v12+-1*v6+-1*v4,2*v13+-1*v7+-1*v5,2*v14+-1*v8+-1*v6,2*v15+-1*v7+-1*v1,
 2*v16+-1*v8+-1*v2,-1+-1*v17*v16^2+-1*v17*v15^2+2*v17*v16*v14+-1*v17*v14^2+2*v17*v15*v13+
 -1*v17*v13^2+v17*v12^2+v17*v11^2+-2*v17*v12*v10+v17*v10^2+-2*v17*v11*v9+v17*v9^2;
 i=subst(i,v1,0,v2,0,v3,0,v4,1);groebner(i)!=1; -> singular
12:56:23.835 DEBUG: geogebra.common.l.q.n.a[-1]: singular -> 0
12:56:23.836 DEBUG: geogebra.common.l.q.i.<init>[-1]: Benchmarking: 47 ms
12:56:23.836 DEBUG: geogebra.common.l.q.i.<init>[-1]: Statement is TRUE
```

4.3 Ceva's theorem

Triangle ABC and its arbitrary internal point D was constructed in Fig. 4. Now intersection points of lines AD, BD, CD and the appropriate sides are E, F,

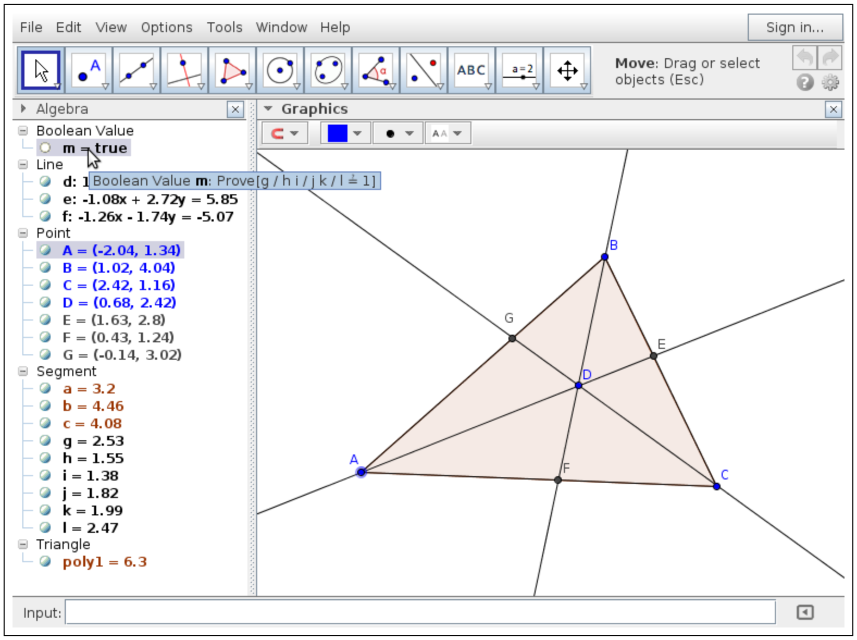

Fig. 4. Ceva's theorem

G, respectively. Now let us define $g = AG$, $h = GB$, $i = BE$, $j = EC$, $k = CF$, $l = FA$, then $g/h \cdot i/j \cdot k/l = 1$. Here OGP's Area method is the only possible way to get a useful ATP answer to decide the statement. To start PP we enter **Prove[g/h i/j k/l==1]**. The result is computed in 4 milliseconds in OGP's area subsystem, but since other methods were also attempted to use, the total time is 95 milliseconds spent in PP:

```
13:22:01.959 DEBUG: geogebra.m.o$a.run[-1]: Using AUTO
13:22:01.959 DEBUG: geogebra.common.p.y.a[-1]: Using RECIOS_PROVER
13:22:01.960 DEBUG: geogebra.common.p.y.a[-1]: Using BOTANAS_PROVER
[...] not fully implemented
13:22:01.966 DEBUG: geogebra.common.p.y.a[-1]: Using OPENGEOPROVER_WU
[...]
13:22:01.987 INFO: a.b.a.a.a[-1]: Reading input geometry problem...
13:22:01.999 DEBUG: a.b.g.a.e.a[-1]: Args before parsing : ((Segment[A, G] / Segment[G, B]
 Segment[B, E] / Segment[E, C]) Segment[C, F] / Segment[F, A]),1
13:22:02.021 ERROR: a.b.a.a.c.e[-1]: Failed to convert statement - missing input argument
13:22:02.022 DEBUG: geogebra.common.p.y.b[-1]: Prover results
13:22:02.022 DEBUG: geogebra.common.p.y.b[-1]: success: false
13:22:02.022 DEBUG: geogebra.common.p.y.b[-1]: failureMessage: Failed in reading input
 geometry theorem
13:22:02.022 DEBUG: geogebra.common.p.y.b[-1]: proverResult: null
13:22:02.023 DEBUG: geogebra.common.p.y.b[-1]: proverMessage: null
13:22:02.023 DEBUG: geogebra.common.p.y.b[-1]: time: null
13:22:02.023 DEBUG: geogebra.common.p.y.b[-1]: numTerms: null
13:22:02.024 DEBUG: geogebra.common.p.y.a[-1]: Using OPENGEOPROVER_AREA
[...]
13:22:02.026 INFO: a.b.a.a.a[-1]: Reading input geometry problem...
13:22:02.030 DEBUG: a.b.g.a.e.a[-1]: Args before parsing : ((Segment[A, G] / Segment[G, B]
 Segment[B, E] / Segment[E, C]) Segment[C, F] / Segment[F, A]),1
13:22:02.037 INFO: a.b.a.a.d.e[-1]: Converting equal statement. Arguments :
```

```
13:22:02.037 INFO: a.b.a.a.d.e[-1]:   ((Segment[A, G] / Segment[G, B] Segment[B, E] /
 Segment[E, C]) Segment[C, F] / Segment[F, A])
13:22:02.037 INFO: a.b.a.a.d.e[-1]:   1
13:22:02.037 ERROR: a.b.a.a.d.e[-1]: A part of the equality is not the label of a known
 construction - trying to parse it as an expression
13:22:02.046 INFO: a.b.a.a.a[-1]: Invoking prover method...
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: Prover results
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: success: true
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: failureMessage: null
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: proverResult: true
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: proverMessage: null
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: time: 0.004
13:22:02.053 DEBUG: geogebra.common.p.y.b[-1]: numTerms: 0
13:22:02.054 DEBUG: geogebra.common.l.q.i.<init>[-1]: Benchmarking: 95 ms
13:22:02.055 DEBUG: geogebra.common.l.q.i.<init>[-1]: Statement is TRUE
```

As seen, (after taking 1 ms in the R-Prover and realizing that it is not helpful) B-Prover cannot process the construction since measuring segments are not yet implemented for it. OGP's Wu's method is also unable to read the information provided by GeoGebra, thus finally OGP's Area method converts the statement into an internal object, and then successfully processes it.

4.4 Thales' circle theorem

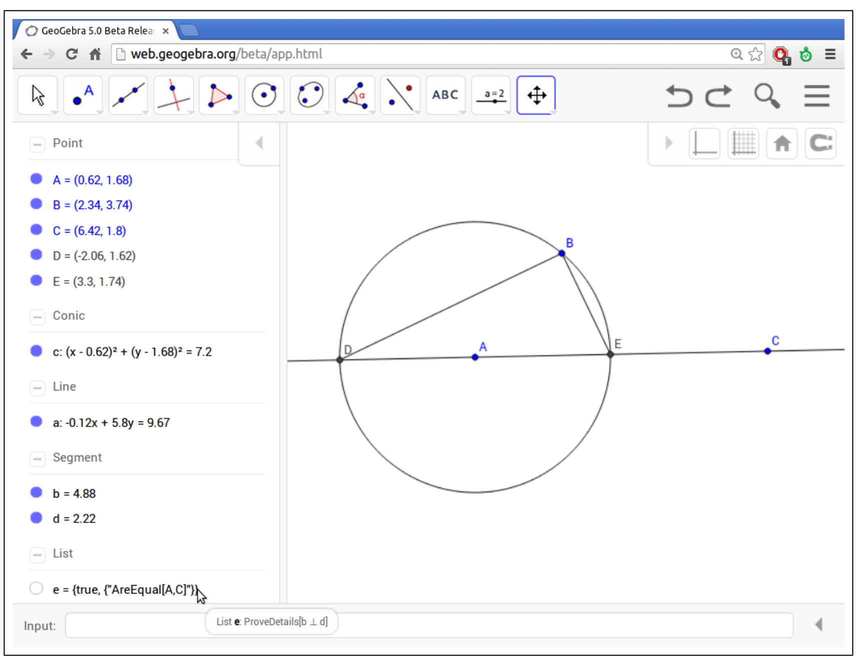

Fig. 5. Thales' circle theorem

Thales' (circle) theorem states that in a given circle with center A, circumpoint B and diameter DE, lines BD and BE are perpendicular. In Fig. 5 we use free points A, B and C (which is a technical point to define D and E as intersection points of line AC and the circle). Let us denote BD by b and BE by d. Now GeoGebra command **ProveDetails[b⊥d]** returns the output list {*true*,{*"AreEqual[A,C]"*}} which has the following meaning:

- the statement "$b \perp d$" holds in general,
- if $A \neq C$ then the statement surely holds.

Clearly, if $A = C$, points D and E are undefined, thus the statement has no meaning. We emphasize here that Fig. 5 is created by running GeoGebra in a web browser. Thus here the only possible method is B-Prover, and only with the embedded CAS, Giac. In this example the construction is loaded from an external file, thus the Javascript version of Giac (giac.js) must be preloaded before any concrete computations in PP. This is the technical reason why we can see that PP ran multiple times (first it reported "undefined" result in 18 ms, then another "undefined" in 3 ms, and the final computation took 2091 ms—"undefined" here is displayed as "`Statement is null`").

```
14:31:25.918 DEBUG: ?: Using AUTO
14:31:25.918 DEBUG: ?: Using BOTANAS_PROVER
14:31:25.919 DEBUG: ?: Testing local CAS connection
14:31:25.928 DEBUG: ?: starting CAS
14:31:25.934 DEBUG: ?: Local CAS evaluates 1 to ?
14:31:25.935 DEBUG: ?: Benchmarking: 18 ms
14:31:25.936 DEBUG: ?: Statement is null
14:31:25.967 DEBUG: ?: Using AUTO
14:31:25.967 DEBUG: ?: Using BOTANAS_PROVER
14:31:25.968 DEBUG: ?: Testing local CAS connection
14:31:25.969 DEBUG: ?: Local CAS evaluates 1 to ?
14:31:25.971 DEBUG: ?: Benchmarking: 3 ms
14:31:25.971 DEBUG: ?: Statement is null
14:31:26.704 DEBUG: ?: giac.js loading success
14:31:27.273 DEBUG: ?: Using AUTO
14:31:27.274 DEBUG: ?: Using BOTANAS_PROVER
14:31:27.275 DEBUG: ?: Testing local CAS connection
14:31:27.275 DEBUG: ?: giac eval: 1
14:31:27.668 DEBUG: ?: giac  input:1
14:31:27.673 DEBUG: ?: giac output:1
14:31:27.675 DEBUG: ?: Local CAS evaluates 1 to 1
[...]
14:31:27.719 DEBUG: ?: Thesis reductio ad absurdum (denied statement):
14:31:27.720 DEBUG: ?: 6. -1+-1*v12*v10*v8+-1*v12*v9*v7+v12*v10*v4+v12*v8*v4+
 -1*v12*v4^2+v12*v9*v3+v12*v7*v3+-1*v12*v3^2
14:31:27.727 DEBUG: ?: Eliminating system in 10 variables (6 dependent)
[...]
14:31:27.817 INFO: ?: [eliminateFactorized] input to cas: [[containsvars(poly,varlist):=
 {local ii; for (ii:=0; ii<size(varlist); ii++) { if (degree(poly,varlist[ii])>0)
 { return true } } return false}],[myeliminate(polylist,varlist):={local ii,jj,kk;
 kk:=[]; jj:=gbasis(polylist,varlist,revlex); for (ii:=0; ii<size(jj); ii++) { if
 (!containsvars(jj[ii],varlist)) { kk:=append(kk,jj[ii]) } } return kk }],[ff:=""],
 [aa:=myeliminate([-1*v7*v6+v8*v5,-1*v8^2+-1*v7^2+v4^2+v3^2,-1*v9*v6+v10*v5,
 -1*v10^2+-1*v9^2+v4^2+v3^2,-1+v11*v10^2+v11*v9^2+-2*v11*v10*v8+v11*v8^2+
 -2*v11*v9*v7+v11*v7^2,-1+-1*v12*v10*v8+-1*v12*v9*v7+v12*v10*v4+v12*v8*v4+-1*v12*v4^2
 +v12*v9*v3+v12*v7*v3+-1*v12*v3^2],[v7,v8,v9,v10,v11,v12])],[bb:=size(aa)],[for ii
 from 0 to bb-1 do ff+=("["+(ii+1)+"]: [1]:  _[1]=1");cc:=factors(aa[ii]);dd:=size(cc);
 for jj from 0 to dd-1 by 2 do ff+=("  _["+(jj/2+2)+"]="+cc[jj]); od; ff+=(" [2]: "+
 cc[1]);for kk from 1 to dd-1 by 2 do ff+=(","+cc[kk]);od;od],ff][6]
14:31:27.819 DEBUG: ?: giac eval: [[containsvars... [...]
[...]
14:31:27.902 DEBUG: ?: giac  input:[[containsvars... [...]
```

```
14:31:29.316 DEBUG: ?: giac output:"[1]: [1]:  _[1]=1  _[2]=v5 [2]: 1,1[2]: [1]:  _[1]=1
 _[2]=v6 [2]: 1,1
14:31:29.318 INFO: ?: [eliminateFactorized] output from cas: [1]: [1]:  [...]
14:31:29.324 DEBUG: ?: Considering NDG 1...
14:31:29.329 DEBUG: ?: Trying to detect polynomial v5
14:31:29.349 DEBUG: ?: v5 means x-equality for [v1, v5]
14:31:29.350 DEBUG: ?: Not better than previous NDG score (Infinity), this is Infinity
14:31:29.350 DEBUG: ?: Considering NDG 2...
14:31:29.351 DEBUG: ?: Trying to detect polynomial v6
14:31:29.363 DEBUG: ?: v6 means y-equality for [v2, v6]
14:31:29.364 DEBUG: ?: Found a better NDG score (0.5) than Infinity
14:31:29.365 DEBUG: ?: Benchmarking: 2091 ms
14:31:29.365 DEBUG: ?: Statement is true
```

In the log we can see that Giac returns the elimination ideal for the Recio-Vélez algorithm in the same way as Singular does.

5 Conclusion and future work

The DGS facility of GeoGebra is already well known in schools, but also ATP functionality will be included in the forthcoming version 5. Its careful design in the user interface and the applied methods using open source technology ensure, in our opinion, a good start to offer useful new ways to teach and learn Euclidean geometry.

There is, however, still lot of work to do. Other methods could be implemented either in GeoGebra or OGP (or in even both), the current methods could be parallelized and further improved. Also the outsourced computations could be included in GeoGebra by using the newest compilers.

A completely open question is how readable proofs should be shown in GeoGebra without confusing the students. The JGEX application has already many promising ways which should be further discussed not only by ATP experts but teachers as well, especially those who teach in secondary schools. After such a consensus we would like to start to implement the **ShowProof** command in GeoGebra.

We believe that GeoGebra (and also OGP) could be a common platform for other ATP experts to join and since the source code is completely open to the public, there is no technological obstacle to integrate the knowledge base into a single application. A long term collaboration with experts from various countries would be fruitful in each classroom since GeoGebra helps each student in his or her native language to understand mathematics even more.

6 Acknowledgments

Ivan Petrović, PhD student of University of Belgrade, primary author of OGP had a great part in making OGP generally work. Gábor Ancsin was the leader of the HTML5 based experiments in the GeoGebra Team. Michael Borcherds and Zbyněk Konečný helped regularly to find and eliminate bugs during the development. Last but not least, Bernard Parisse kindly helped in improving Giac to be even more robust for Gröbner basis computations.

Bruno Buchberger kindly invited the author to present a preliminary version of this paper at the RISC Theorema seminar on 9 April 2014. The slides for that talk are available to download at http://ggb1.idm.jku.at/~kovzol/talks/risc2014-2/.

References

1. Hohenwarter, M.: GeoGebra: Ein Softwaresystem für dynamische Geometrie und Algebra der Ebene. Master's thesis, Paris Lodron University, Salzburg, Austria (2002) (In German.).
2. Narboux, J.: GeoProof, a user interface for formal proofs in geometry. In: Mathematical User-Interfaces Workshop, Electronic Proceedings. Schloss Hagenberg, Linz, Austria (2007) http://www.activemath.org/workshops/MathUI/07/proceedings/Narboux-Geoproof-MathUI07.html.
3. Narboux, J.: A graphical user interface for formal proofs in geometry. Journal of Automated Reasoning **39** (2007) 161–180
4. Botana, F., Valcarce, J.: A dynamic-symbolic interface for geometric theorem discovery. Computers and Education **38** (2002) 21–35
5. Ye, Z., Chou, S.C., Gao, X.S.: 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. Volume 6301. Springer-Verlag (2011) 189–195
6. Wang, D.: Geother 1.1: Handling and proving geometric theorems automatically. In: Automated Deduction in Geometry, volume 2930 of Lecture Notes in Artificial Intelligence, Springer-Verlag (2004) 194–215
7. Petrović, I., Janićič, P.: Integration of OpenGeoProver with GeoGebra (2012) http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/IvanPetrovic.pdf.
8. Quaresma, P., Janičić, P.: GeoThms — a web system for Euclidean constructive geometry. In: Electronic Notes in Theoretical Computer Science. Proceedings of the 7th Workshop on User Interfaces for Theorem Provers (UITP 2006). Volume 174. (2007) 35–48
9. Pham, T.M., Bertot, Y., Narboux, J.: A Coq-based library for interactive and automated theorem proving in plane geometry. In Murgante, B., Gervasi, O., Iglesias, A., Taniar, D., Apduhan, B., eds.: Computational Science and Its Applications – ICCSA 2011. Volume 6785 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2011) 368–383 http://dx.doi.org/10.1007/978-3-642-21898-9_32.
10. Wikipedia: Coq — Wikipedia, the free encyclopedia (2014) http://en.wikipedia.org/w/index.php?title=Coq.
11. Botana, F., Kovács, Z., Weitzhofer, S.: Implementing theorem proving in GeoGebra by using a Singular webservice. In: Proceedings EACA 2012, Libro de Resúmenes del XIII Encuentro de Álgebra Computacional y Aplicaciones. Universidad de Alcalá (2012) 67–70
12. Kovács, S., Recio, T., Weitzhofer, S.: Implementing theorem proving in GeoGebra by exact check of a statement in a bounded number of test cases. In: Proceedings EACA 2012, Libro de Resúmenes del XIII Encuentro de Álgebra Computacional y Aplicaciones. Universidad de Alcalá (2012) 123–126

13. Petrović, I., Kovács, Z., Weitzhofer, S., Hohenwarter, M., Janičić, P.: Extending GeoGebra with automated theorem proving by using OpenGeoProver — presentation at the CADGME 2012 conference in Novi Sad, Serbia. `http://ggb1.idm.jku.at/~kovzol/talks/cadgme12/06/06.pdf` (2012)
14. Ye, Z., Chou, S., Gao, X.: Visually dynamic presentation of proofs in plane geometry, part 1. Basic features and the manual input method. Journal of Automated Reasoning **45** (2010) 213–241
15. Ye, Z., Chou, S., Gao, X.: Visually dynamic presentation of proofs in plane geometry, part 2. Automated generation of visually dynamic presentations with the full-angle method and the deductive database method. Journal of Automated Reasoning **45** (2010) 243–266
16. Kapur, D.: Using Gröbner bases to reason about geometry problems. Journal of Symbolic Computation **2** (1986) 399–408 `http://www.sciencedirect.com/science/article/pii/S0747717186800074`.
17. Wu, W.T.: On the decision problem and the mechanization of theorem proving in elementary geometry. Scientia Sinica **21** (1978) 157–179
18. Janičić, P., Narboux, J., Quaresma, P.: The area method: a recapitulation. Journal of Automated Reasoning **48** (2012) 489–532
19. Chou, S., Gao, X., Zhang, J.: A deductive database approach to automated geometry theorem proving and discovering. J. Automated Reasoning **25** (2000) 219–246
20. Desfontaines, D.: Theorem proving in GeoGebra: Implementing the area method into OpenGeoProver (internship report) (2012) `http://www.eleves.ens.fr/home/desfonta/InternshipReport.pdf`.
21. Baeta, N., Quaresma, P.: The full angle method on the OpenGeoProver. (2013) `http://ceur-ws.org/Vol-1010/paper-08.pdf`, submitted to THedu'13.
22. Weitzhofer, S.: Mechanic proving of theorems in plane geometry. Master's thesis, Johannes Kepler University, Linz, Austria (2013)
23. Recio, T., Vélez, M.: Automatic discovery of theorems in elementary geometry. J. Autom. Reasoning **23** (1999) 63–82
24. Decker, W., Greuel, G.M., Pfister, G., Schönemann, H.: SINGULAR 3-1-6 — A computer algebra system for polynomial computations. (2012) `http://www.singular.uni-kl.de`.
25. Ancsin, G., Hohenwarter, M., Kovács, Z.: GeoGebra goes web. The Electronic Journal of Mathematics and Technology **7** (2013) 412–418
26. Hearn, A.C.: REDUCE User's Manual Version 3.8 (2004) `http://reduce-algebra.com/docs/reduce.pdf`.
27. Parisse, B.: Giac/Xcas, a free computer algebra system (2013) `http://www-fourier.ujf-grenoble.fr/~parisse/giac.html`.
28. Kovács, Z., Parisse, B.: Giac and GeoGebra — improved Gröbner basis computations, Special semester on applications of algebra and number theory, workshop 3 on computer algebra and polynomials. `https://www.ricam.oeaw.ac.at/specsem/specsem2013/workshop3/slides/parisse-kovacs.pdf` (2013)
29. Zakai, A.: Emscripten: An LLVM-to-JavaScript compiler. (2013) `https://github.com/kripken/emscripten/blob/master/docs/paper.pdf?raw=true`.