Jan Długosz University in Częstochowa
Scientific Issues, Mathematics XVI, Częstochowa 2011

COMPUTER PROOFS IN PLANE GEOMETRY

Martin Billich

Faculty of Education, Catholic University in Ružomberok
Hrabovská cesta 1, 034 01 *Ružomberok, Slovak Republic*
e-mail: billich@ku.sk

Abstract. Over the past 25 years highly successful methods for geometry theorem proving have been developed. We will use elementary and understandable examples to show the nature of the techniques for verification of geometric constructions made with interactive geometry environment and for proving geometric statements. In addition to some informations about the WinGCLC software with specific language, we look at the system GeoThms that integrates Automatic Theorem Provers, Dynamic Geometry Tools and a database. The abovementioned system provides an environment suitable for new ways of studying and teaching geometry at different levels.

1. Introduction

Dynamic geometry software (DGS) is the most widely used software for mathematics in education. DGS allows the user to create complex geometric constructions step by step using free objects such as free points, construct new objects depending on the existing ones (for instance, the line passing through two distinct points) and then move the starting points to explore how the whole construction changes. The corresponding figure is updated in real time. There exist a large number of free and commercial software[1] (e.g. Baghera, Cabri, Cinderella, Dr. Geo, Eukleides, WinGCLC, GeoGebra, Geometer's Sketchpad, Geometrix, Geometry Expert (GEX), Geometry Explorer, Géoplan, GeoNext, GeoProof, KGeo, KIG, Non-Euclid, OpenEuclide, WinGeom). Interactive geometry software can help teachers to illustrate abstract concepts in geometry and students may explore and understand the secret of plane geometry on their own. Therefore, DGS systems are used for two activities:

[1] http://en.wikipedia.org/wiki/Dynamic_geometry_software

(1) to help a student to create geometric constructions; (2) to help a student to explore a figure, invent conjectures, and check facts.

From the beginning, various kinds of DGS have been the paradigm of new technologies applied to mathematics education, area where they have found their most applications. Their convenience in the classroom is almost unanimously praised by education experts. However, questions have been raised on the influence or interaction of the use of DGS on the development of the concept of proof in school curricula [2]. Sometimes, formal proofs have been replaced by the construction of a great number of examples of a configuration, what has come to be known as a visual proof.

Geometry is also an important area for automatic theorem proving (ATP), the field of using automated methods for creating mathematical proofs. The exactness and broad theoretical foundation that is present in geometry and the beauty and elegance of geometry make it a wonderful platform for experimentation and testing for new algebraic and other methods.

Several DGS systems with proof-related features can be roughly classified into two categories [5]:

- systems that permit one to build proofs;
- systems that permit one to check facts using an automated theorem prover.

A breakthrough in automated geometry theorem proving (AGTP) is made by Wen-Tsün Wu. Restricting himself to a class of geometry statements of *equality type*, in 1977 Wu introduced a method which can be used to prove quite difficult geometry theorems efficiently. Here we would like to remind that Wu's method cannot deal with theorems involving inequalities.

AGTP has two major lines of research [4, 9]: the synthetic proof style and the algebraic proof style. *Algebraic proof* style methods are based on reducing geometric properties to algebraic properties expressed in terms of Cartesian coordinates. *Synthetic methods* attempt to automate traditional geometry proof methods. The synthetic methods provide traditional (not coordinate-based), human-readable proofs. In both cases (algebraic or synthetic) we claim that the AGTPs can be used in the learning process.

2. WinGCLC software

WinGCLC package is a tool which enables producing geometrical figures (i.e. digital illustrations) on the basis of their formal descriptions. This approach is guided by the idea of formal geometrical constructions. A geometrical construction is a sequence of specific, primitive construction steps (*elementary constructions*). Figure descriptions in WinGCLC are usually made by a list of definitions of several (usually very few) fixed points (defined in terms of Cartesian plane, e.g. by pairs of coordinates) and a list of construction steps based on that points.

WinGCLC uses a specific language for describing figures. The GCLC language consists of the following groups of commands: *definitions*, *basic constructions*, *transformations*, *drawing commands*, *marking and printing commands*, *low level commands*, *Cartesian commands*, *commands for describing animations*, *commands for the geometry theorem prover*. These descriptions are compiled by the processor and can be exported to different output formats. There is an interface which enables simple and interactive use of a range of functionalities, including making animations.

The theorem prover (GCLCprover) built into WinGCLC is based on Chou's algorithm for proving geometry theorems (*area method*, see [1]). This method belongs to the group of synthetic methods. The main idea of the method is to express hypotheses of a theorem using a set of constructive statements, each of them introducing a new point, and to express a conclusion by an equality of expressions in geometric quantities such as *ratio of directed parallel segments* $\overline{AB}/\overline{CD}$ (where $\overline{AB}$ denotes the *signed length*[2] of a segment AB), *signed area* S_{ABC} (the area of a triangle ABC with a sign depending on the order of the vertices A, B and C[3]) and *Pythagoras difference* $P_{ABC} = \overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$ as a generalization of the Pythagoras equality (for details see [8]).

The proof is then based on eliminating (*in reverse order*) the points introduced before, using for that purpose a set of appropriate lemmas. After eliminating all introduced points, the current goal becomes a trivial equality that can be simply tested for validity. At all stages, different expression simplifications are applied to the current goal.

Let us take next elimination lemma and one example:

Lemma 1. Let S_{ABY} be the signed area of a triangle ABY for distinct points A, B and Y. For collinear points Y, U and V it holds

$$S_{ABY} = \frac{\overline{UY}}{\overline{UV}} S_{ABV} + \frac{\overline{YV}}{\overline{UV}} S_{ABU}.$$

Example 1 (of elimination technique). Let Y be a point on a line passing through a given point W and parallel to a line UV, such that $\overline{WY} = r\overline{UV}$,

[2] If we prescribe a direction from A to B as positive, then $\overline{AB} = |AB|$ and $\overline{BA} = -|AB|$.

[3] S_{ABC} is positive if we move along the perimeter of a triangle from the vertex A to B and C anti-clockwise.

where r can be a rational number, a rational expression in geometric quantities, or a variable. Then it holds:

$$S_{ABY} = S_{ABW} + r(S_{ABV} - S_{ABU}).$$

The constructions accepted by GCLCprover are: construction of a line given by two points; an intersection of two lines; the midpoint of a segment; a segment bisector; a line passing through a given point, perpendicular to a given line; a foot from a point to a given line; a line passing through a given point, parallel to a given line; an image of a point in a given translation; an image of a point in a given scaling transformation; a random point on a given line.

Let us consider the triangle area theorem as an example:

Example 2 (Triangle area theorem). Each median divides the triangle into two smaller triangles which have the same area.

Proof (using the method). Let ABC be a triangle, and M be a midpoint of AB. We first translate the goal into its equivalent using the signed area:

$$S_{AMC} = S_{MBC}.$$

The proof is actually to eliminate a point M. Using Example 1, the above equality of signed areas can be reduced to the expressions as follows:

$$S_{AMC} = S_{CAM} = S_{CAA} + \frac{1}{2}(S_{CAB} - S_{CAA}),$$

$$S_{MBC} = S_{BCM} = S_{BCA} + \frac{1}{2}(S_{BCB} - S_{BCA}).$$

The new goal is:

$$\frac{1}{2}S_{CAB} = \frac{1}{2}S_{BCA}.$$

The proof is completed as $S_{CAB} = S_{BCA}$.

We can use WinGCLC to validate the previous statement by describing the construction and proving the property for given three fixed distinct points A, B, C with M being the midpoint of AB. The WinGCLC code for this construction and the corresponding illustration (LaTeX output), are shown in Figure 1. It can be checked (using GCLCprover) that a median CM divides a triangle ABC into two smaller triangles ($\triangle AMC$ and $\triangle MBC$) which have the same area, i.e. $S_{AMC} = S_{MBC}$. This statement can be given in the code of GCLC language by the following line:

```
point  A 10 10
point  B 70 10
point  C 55 35
midpoint  M A B

drawsegment  A B
drawsegment  B C
drawsegment  A C
drawsegment  C M

cmark_b  A
cmark_b  B
cmark_t  C
cmark_b  M
```

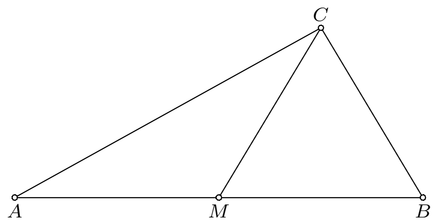

Figure 1: Example 1

```
prove { equal { signed_area3 A M C } { signed_area3 M B C } }
```

The prover produces a short report of information on number of steps performed, on CPU time spent and whether or not the conjecture has been proved. For our example we have:

```
The theorem prover based on the area method used.
 Number of elimination proof steps:  2
 Number of geometric proof steps:    7
 Number of algebraic proof steps:    9
 Total number of proof steps:        18
Time spent by the prover: 0.004 seconds
The conjecture successfully proved.
The prover output is written in the file triangle_area.tex.
```

The prover also generates a proof in LaTeX form (in the file `proof.tex`). We can control the level of details given in the generated proof. The proof consists of *proof steps*. For each step, there is an explanation and its semantic counterpart. This semantic information is calculated for concrete points used in the construction. For our example (in Figure 1), we will get the following:

(1)	$S_{AMC} = S_{MBC}$	, by the statement
(2)	$S_{CAM} = S_{BCM}$	, by geometric simplifications
(3)	$\left(S_{CAA} + \left(\frac{1}{2}(S_{CAB} + (-1 \cdot S_{CAA}))\right)\right) = S_{BCM}$	, by Lemma 29 (M eliminated)
(4)	$\left(0 + \left(\frac{1}{2}(S_{CAB} + (-1 \cdot 0))\right)\right) = S_{BCM}$	, by geometric simplifications
(5)	$\left(\frac{1}{2}S_{CAB}\right) = S_{BCM}$	, by algebraic simplifications
(6)	$\left(\frac{1}{2}S_{CAB}\right) = \left(S_{BCA} + \left(\frac{1}{2}(S_{BCB} + (-1 \cdot S_{BCA}))\right)\right)$	, by Lemma 29 (M eliminated)
(7)	$\left(\frac{1}{2}S_{CAB}\right) = \left(S_{CAB} + \left(\frac{1}{2}(0 + (-1 \cdot S_{CAB}))\right)\right)$	, by geometric simplifications
(8)	$0 = 0$	, by algebraic simplifications

Q.E.D.

3. GeoThms

GeoThms[4], is a framework that links DGS (GCLC and Euklides), AGTP (GCLCprover), and a repository of geometry problems (GeoDB), providing a common web interface for all these tools (see Figure 2).

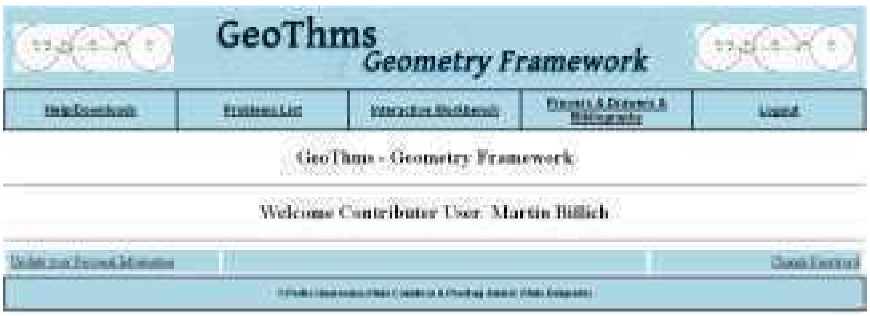

Figure 2: GeoThms – Regular Users Page

Integration of GeoThms with dynamic geometry software and automatic theorem provers and its repository of theorems, figures and proofs give the user the possibility to browse easily through the list of geometric problems, their statements, illustrations and proofs, and also to use interactively the drawing and proving programs (see Figure 3).

[4]GeoThms is a set of PHP scripts of top of a MySQL database and is accessible from http://hilbert.mat.uc.pt/GeoThms.

Figure 3: GeoThms – Theorem Report

As a web service GeoThms emphasizes [6]: (1) a simple interface based on using geometrical specification languages of the underlying geometrical tools; (2) a low communication burden. A basic communication, concerning describing geometrical constructions and conjectures, is based on formal languages of the underlying geometrical tools. Within GeoThms, data are presented in *textual form* as GCLC code, or as XML rendered as HTML, and *graphical form* as JPEG image, or as SVG image. When adding new geometrical tools, it will be sufficient to develop converters from its format to XML and vice versa. This enables converting from any format to any other, and consequently makes usable the whole of the repository to any geometrical tool.

4. Conclusion

In this paper we present some advantages of interactive geometry system WinGCLC, automated theorem prover GCLCprover, and geometry framework GeoThms. The built-in module is based on the area method for Euclidean geometry. The main advantage of this method is that each step of the generated proof has clear geometric meanings and the proofs are generally elegant. The computer program based on the area method has produced proofs of more than 500 geometry theorems, some of which are even shorter than those given by geometry experts. A drawback is that the students must be taught the "area axioms" instead of the standard Euclidean axioms.

Acknowledgements

This work was supported by the grant KEGA 463-035KU-4/2010.

References

[1] Shang-Ching Chou, Xiao-Shan Gao, Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants: Multiple and shortest proof generation. *Journal of Automated Reasoning*, **17**, 325–347, 1996.

[2] J. Escibrano, F. Botana, M.A. Abandés. Adding remote computational capabilities to dynamic geometry systems. *Mathematics and Computers in Simulation*, **80**, 1177–1184, 2010.

[3] P. Janičić, P. Quaresma. System Description: GCLCprover + GeoThms. In: *International Joint Conference on Automated Reasoning. Lecture Notes in Artificial Intelligence*, pp. 145—150, Springer, Berlin 2006.

[4] N. Matsuda, K. Vanlehn. GRAMY: A geometry theorem prover capable of construction. *Journal of Automated Reasoning*, **32**, 3–33, 2004.

[5] J. Narboux. A graphical user interface for formal proofs in geometry. *Journal of Automated Reasoning*, **39**, 161–180, 2007.

[6] P. Quaresma, P. Janičić. GeoThms – a Web system for Euclidean constructive geometry. *Electronic Notes in Theoretical Computer Science*, **174**(2), 35–48, 2007.

[7] P. Quaresma, P. Janičić. Integrating dynamic geometry software, deduction systems, and theorem repositories. In: *Mathematical Knowledge Management. Lecture Notes in Artificial Intelligence*, pp. 280—294, Springer, Berlin 2006.

[8] P. Quaresma, P. Janičić. *The Area Method, Rigorous Proofs of Lemmas in Hilbert's Style Axioms Systems.* Technical Report TR2006/001, Center for Informatics and Systems of the University of Coimbra, 2009.

[9] V. Santos, P. Quaresma. Adaptative learning environment for geometry. In: *Advances in Learning Processes*, M.B. Rosson (Ed.), pp. 71–91, InTech, 2010.