South Bohemia Mathematical Letters
Volume 18, (2010), No. 1, 39-46.

DEDUCTION AND PROVING OF GEOMETRIC STATEMENTS IN INTERACTIVE GEOMETRY ENVIRONMENT

MARTIN BILLICH

Abstract. Synthetic geometry has been making a comeback to university education, thanks to modern applications of geometry software tools. In this paper we look at an application of automated theorem proving (ATP) in the verification of constructions made with interactive geometry environment. In addition to basic information about the WinGCLC software, the paper offers some examples of using specific ATP system, GCLCprover. Using this tool, we can efficiently prove many complex geometric problems and theorems.

Introduction

The goal of the mathematics teacher is for her/his students to engage in advanced mathematical thinking at any grade level. Building conjectures in geometry that lead to formal and informal proof is one example of advanced mathematical thinking.

There is a range of geometry software tools, covering different geometries and geometry problems. The Dynamic Geometry Softwares (DGS) visualize geometric objects and link formal, axiomatic nature of geometry with its standard models (e.g., Cartesian model) and corresponding illustrations. Typically, the user starts a construction with several points, construct new objects depending on the existing ones, and then move the starting points to explore how the whole construction changes. Dynamic geometry software can help teachers to illustrate abstract concepts in geometry. Students may explore and understand the secret of Euclidean geometry on their own. They may see with their own eyes that this or that geometrical theorem is true. To prove a given result using a DGS we would have to prove that the result is true in every possible configurations.

In this paper we present basic information of WinGCLC - tool for producing geometrical illustrations for LaTeX, and automated theorem prover, GCLCprover. All constructions and conjectures are stored in formal, declarative representation that can be used as a description of a construction, a description of a figure, and also as a formal description of a conjecture that can be attempted to be proved by the developed theorem prover.

Received by the editors 11th November 2010.
1991 *Mathematics Subject Classification.* 68T15, 51A99.
Key words and phrases. deductive-check system, automated geometry theorem proving, area method, GCLC software, geometry constructions.
The paper was supported by the project KEGA 463-035KU-4/2010.

1. Background

In this section we give some basic background information about formal constructions with WinGCLC software, and an application GCLCprover. For more details about WinGCLC system, see [2,4,6].

1.1. **WinGCLC.** 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*).

WinGCLC uses a specific language for describing figures. GC 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.

While a construction is an abstract procedure, in order to make its representation in Cartesian plane, we still have to make a some link between these two. For instance, given three vertices of a triangle we can construct a center of its inscribed circle (by using primitive constructions), but in order to represent this construction in Cartesian plane, we have to take three particular Cartesian points as vertices of the triangle. Thus, figure descriptions in WinGCLC are usually made by a list of definitions of several (usually very few) fixed points and a list of construction steps based on that points.

1.2. **GCLCprover.** Automated theorem proving in geometry has two major lines of research: *synthetic proof style* and *algebraic proof style*. Algebraic proof style methods are based on reducing geometric properties to algebraic properties expressed in terms of Cartesian coordinates. These methods are usually very efficient, but the proofs they produce do not reflect the geometric nature of the problem and they give only a *yes* or *no* conclusion. Synthetic methods attempt to automate traditional geometry proof methods.

The geometry theorem prover built into WinGCLC is based on the *area method* (see [7,8]). 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 segments*, *signed area* and *Pythagoras difference*.

Definition 1.1. For four collinear points A, B, C and D such that $C \neq D$, the ratio of directed parallel segments, denoted $\frac{\overrightarrow{AB}}{\overrightarrow{CD}}$ is a real number. If P and Q are points such that $CDPQ$ is a parallelogram and A, B are on the line PQ, then

$$\frac{\overrightarrow{AB}}{\overrightarrow{CD}} = \frac{\overrightarrow{AB}}{\overrightarrow{QP}}.$$

Definition 1.2. The signed area of triangle ABC, denoted S_{ABC}, is the area of the triangle with a sign depending on its orientation in the plane.[1]

[1] We have anticlockwise, positive sign, and clockwise, negative sign.

Definition 1.3. The signed area of a quadrilateral $ABCD$ is defined as $S_{ABCD} = S_{ABC} + S_{ACD}$.

The Pythagoras difference is a generalization of the Pythagoras equality regarding the three sides of a right triangle, to an expression applicable to any triangle.

Definition 1.4. For three points A, B and C, the Pythagoras difference, denoted P_{ABC}, is defined in the following way: $P_{ABC} = AB^2 + CB^2 - AC^2$.

Definition 1.5. For a quadrilateral $ABCD$, the Pythagoras difference P_{ABCD} is defined as $P_{ABCD} = P_{ABD} - P_{CBD}$.

Expressing some common geometric notions using S_{ABC}, ratios and P_{ABC} are given in Table 1.

Geometric notions	**Formalizations**
Points A and B are identical	$P_{ABA} = 0$
Points A, B, C are collinear	$S_{ABC} = 0$
AB is perpendicular to CD	$P_{ACD} = P_{BCD}$
AB is parallel to CD	$S_{ACD} = S_{BCD}$
M is the midpoint of AB	$\frac{\overrightarrow{AM}}{\overrightarrow{MB}} = 1$
AB has the same length as CD	$P_{ABA} = P_{CDC}$

Table 1.

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

Here we present only two elimination lemmas that are the base for the area method's algorithm (see, for instance, [7] for a survay).

Lemma 1.6 (The Co-side Theorem)**.** *Let M be the intersection of two non-parallel lines AB and PQ and $Q \neq M$ (Figure 1). Then it holds that*

$$\frac{\overrightarrow{PM}}{\overrightarrow{QM}} = \frac{S_{PAB}}{S_{QAB}}; \quad \frac{\overrightarrow{PM}}{\overrightarrow{PQ}} = \frac{S_{PAB}}{S_{PAQB}}; \quad \frac{\overrightarrow{QM}}{\overrightarrow{PQ}} = \frac{S_{QAB}}{S_{PAQB}}.$$

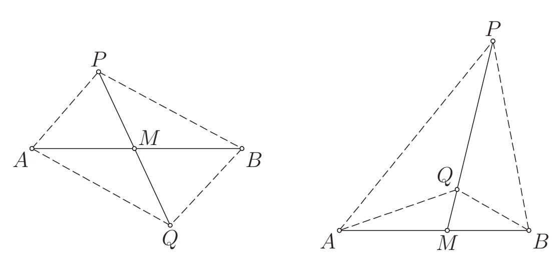

Figure 1. Lemma 1.6

Lemma 1.7. *Let $G(Y)$ be one of the following geometric quantities: S_{ABY}, S_{ABCY}, P_{ABY}, P_{ABCY} for distinct points A, B, C and Y. For collinear points Y, U and V it holds*

$$G(Y) = \frac{\overrightarrow{UY}}{\overrightarrow{UV}} G(V) + \frac{\overrightarrow{YV}}{\overrightarrow{UV}} G(U).$$

Example (of elimination technique). Let Y be the foot of the perpendicular constructed from a point P to a line UV. Then, Y can be eliminated from S_{ABY} by using:

$$S_{ABY} = \begin{cases} S_{ABU} & \text{if } AB \parallel UV \\ S_{ABP} & \text{if } AB \perp UV \\ \frac{S_{ABV} P_{PUAV}}{P_{UVU}} & \text{if } U, V \text{ and } A \text{ are collinear} \\ \frac{S_{AUV} S_{PUBV}}{P_{UVU}} & \text{if } U, V \text{ and } B \text{ are collinear} \end{cases}$$

The theorem prover can prove *any* geometry theorem expressed in terms of geometry quantities, and involving only points introduced by using the commands `point`, `line`, `intersec`, `midpoint`, `med`, `perp`, `foot`, `parallel`, `translate`, `towards`, `online` (see [4,7]).

1.3. **Geometric Constructions.** The area method is used for proving constructive geometry conjectures: statements about properties of objects constructed by some fixed set of elementary constructions, which have a specific form:

$$(C_1, C_2, \ldots, C_m; G)$$

where C_i, for $1 \leq i \leq m$, are elementary construction steps, and the conclusion of the statements, G is of the form $E_1 = E_2$, where E_1 and E_2 are polynomials in geometric quantities of the points introduced by the steps C_i. In each of C_i, the points used in the construction steps must be already introduced by the preceding construction.

WinGCLC have a sort of dual view of a given geometrical construction, a formal language describing it, and a graphical interface where the construction given by the formal description is draw. In our DGS we may consider three type of construction errors:

- *syntactic errors* - this type of error is easily detected by GCLC processor. In our case this are the least important, an easily correctable errors.
- *semantic errors* - situations when, for a given concrete set of geometrical objects, a construction is not possible. For instance, two identical points do not determine a line, this type of error will be dealt for a given fixed set of points.
- *deductive errors* - a construction geometrically incorrect, e.g., the intersection of two parallel lines in Euclidean geometry. The formal proof of the correctness (or not) of a given construction can be dealt by GCLCprover integrated with GCLC/WinGCLC software.

2. Examples

In geometry, we often come across the word *median* while studying triangles. Median of a *triangle* is a line segment joining a vertex of a triangle to the midpoint of the opposite side. There are some basic properties of medians which make them very important in mathematics. They are as follows:

Example 1. (*Concurrency of Medians of a Triangle*). Show that the three medians of any triangle are concurrent.

We have to define three points A, B, C and from this ones all other objects should be constructed, namely the midpoints A_1, B_1 and C_1. Let T_1 and T_2 be the pairwise intersections of the medians of triangle ABC. The WinGCLC code for this construction and the corresponding illustration (LaTeX output), are shown in Figure 2.

```
point A 10 10
point B 40 10
point C 30 40

midpoint A_1 B C
midpoint B_1 A C
midpoint C_1 A B

intersec T_1 A_1 A B_1 B
intersec T_2 A_1 A C_1 C

drawsegment A B
drawsegment A C
drawsegment B C

cmark_lb A
cmark_rb B
cmark_t C

cmark_b C_1
cmark_l B_1
cmark_r A_1

cmark_lb T_1
cmark_rb T_2
```

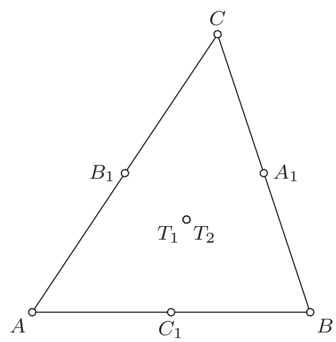

FIGURE 2. Example 1

If we attempt to construct a line T_1T_2 (by adding `line p T_1 T_2` at the end of the code given in Figure 2), we will get the following message (*semantic-based*):

```
Error 11: Run-time error: Bad definition.
Cannot determine line. (Line: 24, position: 6)
File not processed.
```

This information is true for the given particular points A, B, C, i.e., for these three points, the points T_1 and T_2 are identical.

However, if our deductive-check system is turned on, we will also get additional, much deeper information:

```
Deduction check invoked: the property that led to the error
will be tested for validity.

Time spent by the prover: 0.092 seconds
The conjecture successfully proved - the critical property always holds.
The prover output is written in the file error-proof.tex.
```

This means that it was proved that points T_1 and T_2 are always identical, so the construction of a line p determined by these two points is not possible, i.e., the three medians of any triangle are concurrent. This fact we can also stated, using our ATP system, within the command

```
prove { identical T_1 T_2 }
```

or via the geometric quantity *Pythagoras difference* in the following way:

```
prove { equal { pythagoras_differnce3 T_1 T_2 T_1 } 0 }.
```

Example 2. The common point of the three medians of a triangle divides each in the ratio 2:1 (with the larger portion toward the vertex and the smaller portion toward the side).

Let ABC be a triangle, and let A_1 and B_1 be the midpoints of BC and AC respectively. The crucial point is: do the medians AA_1 and BB_1 intersect at point T which divides medians into a $2:1$ ratio?

We can use WinGCLC to answer this question, by describing the construction and proving the property: given three fixed distinct points A, B, C; let A_1 and B_1 be the midpoints of BC and AC respectively. We can construct a point T as the intersection point of the line segments BB_1 and AA_1. The WinGCLC code for this construction and the corresponding illustration, are shown in Figure 3.

```
point A 10 10
point B 40 10
point C 30 40

midpoint A_1 B C
midpoint B_1 A C
midpoint C_1 A B
intersec T A_1 A B_1 B

drawsegment A B
drawsegment A C
drawsegment B C
drawsegment A A_1
drawsegment B B_1
drawdashsegment C C_1

cmark_b C_1
cmark_l B_1
cmark_r A_1
cmark_r T
cmark_lb A
cmark_rb B
cmark_t C

prove {equal {sratio A T T A_1}{2}}
```

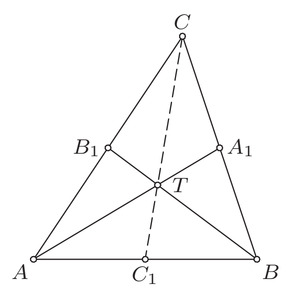

FIGURE 3. Example 2

It can be checked (using GCLCprover) that a point T divides medians AA_1, BB_1 into a $1:2$ ratio, i.e. $\overrightarrow{AT} : \overrightarrow{TA_1} = 2$. This statement can be given in the code of GCLC language by following line:

```
prove {equal {sratio A T T A_1 }{2}}
```

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:   5
 Number of geometric proof steps:    14
 Number of algebraic proof steps:    25
 Total number of proof steps:        44

Time spent by the prover: 0.001 seconds.
The conjecture successfully proved.
The prover output is written in the file medians_proof.tex.
```

The prover generate also proof in LaTeX form (file `medians_proof.tex`) or/and in XML format. We can control the level of details given in 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 (on Figure 3.), we will get the following report (a part of the proof):

(1)	$\frac{\overrightarrow{AT}}{\overrightarrow{TA_1}} = 2$	by the statement
(2)	$\left(-1 \cdot \frac{\overrightarrow{AT}}{\overrightarrow{A_1T}}\right) = 2$	by geometric simplifications
(3)	$\left(-1 \cdot \frac{S_{AB_1B}}{S_{A_1B_1B}}\right) = 2$	by Lemma 8 point T eliminated
⋮	⋮	⋮
(16)	$\frac{\left(-\frac{1}{2} \cdot S_{BAC}\right)}{\left(\left(-\frac{1}{4} \cdot S_{BAC}\right) + \left(\frac{1}{2} \cdot \left(0 + \left(\frac{1}{2} \cdot (0 + (-1 \cdot 0))\right)\right)\right)\right)} = 2$	by geometric simplifications
(17)	$\frac{-\frac{1}{2}}{-\frac{1}{4}} = 2$	by algebraic simplifications
(18)	$-\frac{1}{2} = -\frac{1}{2}$	by algebraic simplifications

Q.E.D.

At the end of the main proof, all non-degenerative conditions are listed:

- $S_{A_1B_1B} \neq S_{AB_1B}$ i.e., lines A_1A and B_1B are not parallel (construction based assumption)
- $P_{TA_1T} \neq 0$ i.e., points T and A_1 are not identical (conjecture based assumption)

3. Conclusion

In this paper we gave some advantages of interactive geometry system WinGCLC and automated theorem prover GCLCprover. The built-in deduction module is based on the area method for Euclidean geometry. This method produced proofs that are human-readable, and can efficiently prove many non-trivial geometric statements and theorems. This complex system for constructive geometry provides an environment for modern ways of studying and teaching geometry at different levels.

Acknowledgment

An essential part of this article has been presented at 30th Conference on Geometry and Graphics, and published in Proceedings of this conference ([9]).

References

[1] Chou, Shang-Ching - Gao, Xiao-Shan - Zhang, Jing-Zhong: *Automated production of traditional proofs for constructive geometry theorems*. In Moshe Vardi, editor, Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science LICS, IEEE Computer Society Press, 48–56, 1993.

[2] Janičić, P.: *GCLC a tool for constructive Euclidean geometry and more than that*. In: Proceedings of International Congress of Mathematical Software, Lecture Notes in Computer Science, vol. 4151, Springer Berlin, 58–73, 2006.

[3] Janičić, P.: *Geometry Constructions Language*. Journal of Automated Reasoning, Volume 44, Numbers 1-2, Springer Netherlands, 3–24, 2010.

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

[5] Janičić, P. - Quaresma, P.: *Automatic verification of regular constructions in dynamic geometry systems*. In: F. Botana, T. Recio (eds.) Proceedings of Automated Deduction in Geometry (ADG06), Lecture Notes in Artificial Intelligence, vol. 4869, 39-51, Springer-Verlag, Pontevedra, Spain 2007.

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

[7] Quaresma, P. - Janičić, P.: *Framework for the constructive geometry*. Technical Report TR2006/001, Center for Informatics and Systems of the University of Coimbra, 2006.

[8] Quaresma, P. - Janičić, P.: *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] Billich, M.: *Shape Deduction and proving of geometric statements in interactive geometry environment*, Proceedings of 30th Conference on Geometry and Graphics, 2010, 33–42, (available on http://www.csgg.cz/30zlenice/sbornik2010.pdf).

Faculty of Education, Catholic University in Ružomberok, Hrabovská cesta 1, 034 01 Ružomberok, Slovakia

E-mail address: billich@ku.sk