ACM Communications in Computer Algebra, Vol. 50, No. 3, Issue 197, September 2016

ISSAC 2016 Software Presentations

Communicated by Roman Pearce

Distinguished Software Demonstration Award

Development of automatic reasoning tools in GeoGebra

Miguel Abánades*
Universidad Rey Juan Carlos, Spain
`miguelangel.abanades@urjc.es`

Francisco Botana*
Universidad de Vigo, Spain
`fbotana@uvigo.es`

Zoltán Kovács
The Private University College of Education of the Diocese of Linz, Austria
`zoltan@geogebra.org`

Tomás Recio*
Universidad de Cantabria, Spain
`tomas.recio@unican.es`

Csilla Sólyom-Gecse
Babes-Bolyai University, Romania
`solyomcsilla@yahoo.com`

Abstract

Much effort has been put into the implementation of automatic proving in interactive geometric environments (e.g. Java Geometry Expert, GeoGebra). The closely related concept of automatic discovery, remains however almost unexplored.

This software presentation will demonstrate our results towards the incorporation of automatic discovery capabilities into GeoGebra, an educational software with tens of millions of users worldwide. As main result, we report on a new command, currently available in the official version, that allows the automatic discovery of loci of points in diagrams defined by implicit conditions. This represents an extension of a previous command, similar in nature, but restricted to loci defined by the standard mover-tracer construction. Our proposal successfully automates the 'dummy locus dragging' in dynamic geometry. This makes the cycle conjecturing-checking-proving accessible for general users in elementary geometry.

1 Introduction

In this software presentation, we will show the implementation of an extended version of the `LocusEquation` GeoGebra command; extension that is available in this dynamic geometry system since version 5.0.213.0, dated March 12, 2016. This presentation will demonstrate how this extension makes possible the automatic discovery of elementary geometry statements within a free dynamic software used by tens of millions of people worldwide.

That the `LocusEquation` GeoGebra command is ready for use under real conditions (i.e. school context) is supported by the fact that there are already examples of its use by teachers[1], even though the command has only been available to the general users for a few weeks.

---

*Partially supported by the Spanish Ministerio de Economía y Competitividad and by the European Regional Development Fund (ERDF), under the Project MTM2014–54141–P.

[1]E.g. `http://tube.geogebra.org/m/DCSFzaph`

2 Automatic deduction in geometry

By automatic proving of elementary geometry theorems, we refer to the theorem proving approach via computational algebraic geometry methods, as initiated by Wu [8] forty years ago, and popularized by the book of Chou [2]. Roughly speaking, the idea is to provide algorithms, using computer algebra methods, for confirming (or refuting) the truth of some given geometric statement. More precisely, the goal is to decide whether a given statement is *generally* true or not, i.e. true except for some degenerate cases, to be described by the algorithm. See [2] for an early collection of examples of highly non trivial theorems in elementary geometry successfully verified by a variety of symbolic computation methods.

Roughly speaking this approach proceeds by translating geometric facts, say hypotheses H and theses T, into systems of polynomial equations, say S_H, S_T, and then, translating geometric statements ($H \Rightarrow T$) as inclusion tests $S_H \subseteq S_T$ between the solution sets of the corresponding systems of equations. Such inclusion tests are then elucidated by some computer algebra tools deciding if a polynomial f is or not an algebraic combination of some given collection of polynomials S, which is—approximately—a way to check whether the roots of f form a superset or not of the solution set of the system $S = 0$.

A closely related, albeit different, issue is that of the automatic *discovery* of theorems, an issue already present in the pioneer work of Wang [7] and Kapur [4] (see [6] for a large collection of references on this topic). Roughly speaking, automatic *proving* deals with establishing whether some statement holds true in most instances, while automatic *discovery* –in its most general conception– addresses the case of statements $H \Rightarrow T$ that are false in most relevant cases. In fact, it aims to automatically produce additional, necessary, hypotheses H' for the (new) statement $(H \wedge H') \Rightarrow T$ to be true. One must remark that the search for complementary hypotheses has to be done in terms of the free variables for the construction.

Describing the *implicit* geometric locus of a point subject to some geometric constraints, say finding the locus set of points P for which its projection onto the three sides of a given triangle form a triangle of given constant area ([2], Chapter IV, Example 5.8) can be considered as a variant of this 'automatic discovery' approach. Indeed, the steps in the construction of the projections of P can be considered as the hypotheses H, while the given constraints over the point P (e.g. requiring that the area of the triangle described by the three projections of P over the sides of the given triangle must be constant) can be considered as the proposed thesis T, one that is false for arbitrary positions of P; finally, the description H' of the locus (for point P to verify that its three projections form a triangle of fixed area) can be understood as the extra, necessary hypotheses required for the given statement to hold true, so that $H \wedge H' \Rightarrow T$.

While automatic proving using computer algebra methods has been used in dynamic geometric software for years[2], similar automatic discovery abilities are not present in software ready for universal use. Following our goal towards the popularization of tools for automatic reasoning in geometry, we have collaborated with other authors providing automatic proving resources to GeoGebra [1]. Continuing this trend, this software presentation showcases our recent work on the addition of discovery capabilities to GeoGebra based on the computational approach described in [5, 3].

3 The GeoGebra command `LocusEquation`

Automatic discovery in GeoGebra requires that the user first constructs a geometric diagram. Although theoretically all *algebraic* constructions (i.e. those composed of elements that can be expressed by polynomial equations) can serve as initial data for GeoGebra's discovery tool, technical reasons, mainly related to computational time limitations, restrict the applicability of the tool for some involved constructions. Moreover, note that non-algebraic elements, such as the graph of a sine function, fall out of the scope of the method, algebraic in nature.

[2]See Java Geometry Expert, JGEX, `http://www.cs.wichita.edu/~ye/gex.html`, for a paradigmatic example.

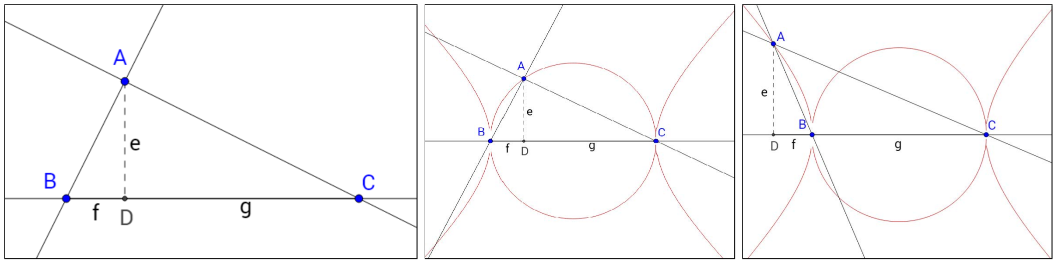

Figure 1: Right triangle altitude theorem: $e^2 = f \cdot g$.

After constructing a geometric diagram the user needs to type the command `LocusEquation` with two parameters: the sought thesis T (which must be an atomic Boolean expression) and a free point P 'supporting' the discovery[3]. The Boolean expression plays the role of the *extra condition* that we require our diagram to satisfy. The free point P is the point over which the sought extra hypothesis will be stated; i.e., in algebraic terms, the symbolic coordinates of P will be the only variables of the polynomials conforming the necessary conditions obtained as a result of the discovery process. As a result, `LocusEquation[T,P]` will produce a set V (providing its implicit equation) such that 'if T is true then $P \in V$'. It should be noted that the basic points of the construction—other than P—are fixed, that is, their numerical coordinates will be used in the discovery computation. Recall that P has always symbolic coordinates. Thus, we are discovering on a specific instance of the general construction, so it could happen that we discover some incidental property only related to this particular instance. For example, if we intend to construct a general triangle but we actually draw—without noticing it—an isosceles one, we can find out statements which are true just for this particular kind of triangles. Of course, this confusion will be clarified in the *proving* phase, when intending to check the validity, in general, of the obtained result.

As a simple illustrative example, let us 'discover' a generalization of the right triangle altitude theorem, which says that, for a right triangle, we have $e^2 = f \cdot g$ where e is the length of the altitude drawn from the vertex of the right angle to its hypotenuse, and f and g are the lengths of the segments BD and CD, as described in Fig. 1, left. For what positions of the vertex A is this equation still true?

As explained above, we obtain the answer by typing `LocusEquation[e*e == f*g, A]`. For these particular vertices $B(-1, 0)$ and $C(1, 0)$ we get the algebraic curve given by the polynomial $y^4+2x^2-x^2-1$, that factors as the product $(-x^2+y^2+1)\cdot(x^2+y^2-1)$. So we see that the locus set includes the expected circle with diameter BC corresponding to right triangles ABC, but also the hyperbola $y^2 - x^2 = -1$. Figure 1 (right) shows a non-right triangle satisfying the right triangle altitude theorem relation.

Let us insist on the fact that points B, C are not generally considered, but their numerical coordinates are used in the computations. This implementation decision tries to imitate the traditional inductive process, reasoning over on concrete situations as a first step.

Once we obtain the equation of the hyperbola as a necessary condition that A has to satisfy if $e^2 = f \cdot g$, we can proceed to prove its sufficiency, namely that $e^2 = f \cdot g$ if $A(x, y)$ is a point in the hyperbola $y^2 - x^2 = -1$. We have that e is given by the y-coordinate of A (or its opposite), and therefore $e^2 = x^2 - 1$. Also we have that $f = distance(D, B) = \sqrt{(x + 1)^2}$ and $g = distance(D, C) = \sqrt{(x - 1)^2}$, so $f \cdot g = x^2 - 1$, which finally proves that $e^2 = f \cdot g$ and hence the condition is also sufficient.

Although the computations above have been made for the particular instance determined by the points $B(-1, 0)$ and $C(1, 0)$, taking into account that the general result is invariant under dilations (homotheties and translations), we can assert that the necessary condition is also sufficient for two general points B and C. In summary, we have 'discovered' the following generalization of the right triangle altitude theorem:

[3]See documentation in `https://www.geogebra.org/manual/en/LocusEquation_Command`

given any two points B, C, a third point A satisfies that $distance(A, D)^2 = distance(D, B) \cdot distance(D, C)$, where D is the orthogonal projection of A onto the line BC, if and only if A belongs to the circle with diameter BC (i.e. ABC is a right triangle) or to a hyperbola as illustrated by Fig. 1.

References

[1] F. Botana, M. Hohenwarter, P. Janičić, Z. Kovács, I. Petrović, T. Recio, S. Weitzhofer: Automated theorem proving in GeoGebra: Current achievements, *Journal of Automated Reasoning* 55: pp. 39–59, 2015.

[2] S.-C. Chou: 'Mechanical Geometry Theorem Proving', in Mathematics and its Applications, D. Reidel Publ. Comp. Dordrecht, 1988.

[3] G. Dalzotto and T. Recio: On protocols for the automated discovery of theorems in elementary geometry, *Journal of Automated Reasoning* 43: pp. 203–236, 2009.

[4] D. Kapur: Wu's method and its application to perspective viewing. In: Geometric Reasoning. D. Kapur and J.L. Mundy, eds. The MIT press, Cambridge, Ma., 1989.

[5] T. Recio, M.P. Vélez: Automatic discovery of theorems in elementary geometry, *Journal of Automated Reasoning* 23: pp. 63–82, 1999.

[6] T. Recio, M.P. Vélez: 'An introduction to automated discovery in geometry through symbolic computation', in Numerical and Symbolic Scientific Computing: Progress and Prospects, (U. Langer and P. Paule, eds.), Springer Series: Texts and Monographs in Symbolic Computation, Vienna, pp. 25–271, 2011.

[7] D. Wang: A new theorem discovered by computer prover, *Journal of Geometry* 36: pp. 173–182, 1989.

[8] W. Wen-Tsün: 'On the decision problem and the mechanization of theorem-proving in elementary geometry', in Automated theorem proving: After 25 years (W. W. Bledsoe and D. W. Loveland, eds.), AMS, Providence, pp. 213–234, 1984.