

Evolution of Automated Deduction and Dynamic Constructions in Geometry



Pedro Quaresma

1 Introduction

Logic appears in a *sacred* and in a *profane* form. The sacred form is dominant in proof theory, the profane form in model theory

D. van Dalen, Logic and Structure (van Dalen, 1980)

Sacred Form

It can be said that the sacred form began circa 300 BC with the writing of Euclid's Elements. The Elements can be seen as a seminal work, establishing the basis for proof theory, with a collection of definitions, postulates, propositions (theorems and constructions) and mathematical proofs of the propositions. For centuries, it was included in the curriculum of the majority of the universities. For example, in 1692 Tirso de Molina, Superior General of the Jesuit Order wrote a letter with very specific orders to improve the teaching of Mathematics at the Portuguese province (i.e. Universities of Coimbra and Évora). One of the suggestions was the reproduction of the figures in the Elements in such a way that all the students could see those figures and discuss about the geometric properties behind those figures. The combination of that letter with the Portuguese tradition of tiling (“azulejos”) gave rise to a collection of tiles with faithful representation of many of the figures in the *Elementa Geometriae*

This work is funded by national funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project CISUC—UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020.

P. Quaresma (✉)

CISUC, Mathematics Department, Universidade de Coimbra, Coimbra, Portugal
e-mail: pedro@mat.uc.pt

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2022
P. R. Richard et al. (eds.), *Mathematics Education in the Age of Artificial Intelligence*,
Mathematics Education in the Digital Era 17,
https://doi.org/10.1007/978-3-030-86909-0_1

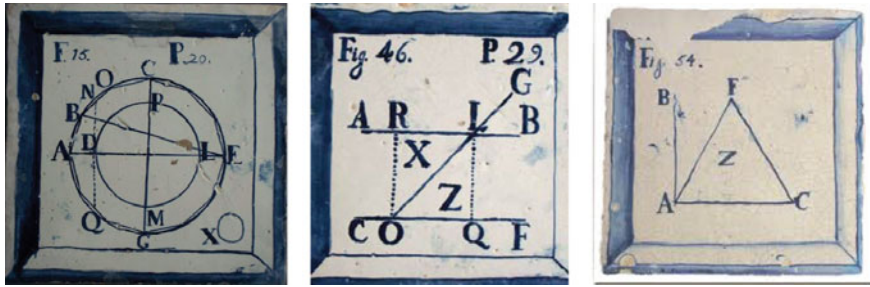


Fig. 1 Azulejos with figures from *Elementa Geometriæ* by André Tacquet

by the Jesuit priest André Tacquet (1612–1660) (see Fig. 1) (Gabriel Silva, 2017; Simões, 2007).¹

David Hilbert in his book *Grundlagen der Geometrie* (1899) established the foundation for a modern axiomatic treatment of Euclidean geometry (Hilbert, 1977). Later, Alfred Tarski built a decision method for elementary algebra and geometry (Tarski, 1951) allowing constructive and even automated approaches for geometry (Beeson, 2015; Quaife, 1989) and, more recently, (1995) Jan von Plato proposed another constructive approach to geometry (von Plato, 1995).

From a computational perspective, the history of geometry automated theorem provers (GATP) began with the early computers and the birth of Artificial Intelligence, in the 1960s. The different sets of axioms of Euclidean Geometry attracted researchers to an attempt to implement synthetic methods, such as the approaches by Gelernter (1995, 1960), Nevis (1975), Elcock, Greeno et al. (1977), Coelho and Pereira (1979, 1986), Chou et al. (1993, 1995). The difficulties found with the synthetic methods, where the need to find a suitable rule to be applied lead to a combinatorial explosion regarding all the possible choices. This resulted in the exploration of other approaches, algebraic, semi-synthetic and logical approaches.

The algebraic style approach is characterized by the translation of geometric problems to algebraic problems, and subsequent development of the proof by the application of algebraic manipulations. The characteristic set method, also known as Wu's method (Chou, 1985; Wu, 1984), the elimination method (Wang, 1995), the Gröbner bases method (Kapur, 1986a, b) and the Clifford algebra approach (Li, 2000) are examples of practical methods of this type. The algebraic approach led to efficient implementations, but, given that all the proofs are developed by algebraic means, the geometric meaning is lost, i.e. apart from a yes/no answer, it is not possible to have a correspondent geometric proof where the axioms of geometry are used. This led to the development of methods capable of, at least partially, combine the geometric readability of synthetics methods with the efficiency of algebraic methods.

¹ Unfortunately most of the tiles were lost after the expelling of the Jesuits from Portugal by the Marquis of Pombal in 1759, the subsequent reform of the University of Coimbra and the construction of new buildings on the expense of the old ones.

The semi-synthetic methods use a set of specific *geometric quantities*, e.g. the *ratio of parallel directed segments* and *signed area*, to build an axiom system where the geometric relations and properties can be represented and the proofs developed using a set of geometric lemmas and simple algebraic manipulations. Examples of such methods are the area method (Chou et al., 1996a; Janičić et al., 2012) and the full-angle method (Chou et al., 1996b). These methods combine the readability of synthetic methods and the efficiency of algebraic methods, being able to prove many geometric theorems, efficiently and with geometric, readable, proofs.

More recently (2000–till present), new synthetic approaches are being proposed; the geometric deductive database method combines the full-angle axiom system with the techniques of deductive databases to develop an efficient GATP capable to prove a large set of geometric problems (Chou et al., 2000). Also tutorial systems like the *QED-Tutrix* (Gagnon et al., 2017; Tessier-Baillargeon et al., 2017) are proposed to address the problem in a more contained form, i.e. instead of trying to implement a generic GATP, the goal is to have an efficient and capable of readable geometric proofs GATP, to specific areas of geometry.

Also to be considered are the logical approaches, like the quantifier elimination method of Tarski (Collins, 1975; Tarski, 1951), or the use of axiom systems for geometry (e.g. Tarski, Quaife (1989)) and then using generic automated theorem provers (ATP) to develop the proofs. Many efficient and capable of proving many geometric conjectures ATP are available, but, like in the algebraic approach, the proof has no correspondence with any form of geometric reasoning. From the view point of a geometer, it is difficult to follow (geometrically) the formal proofs produced by the ATP.

Profane Form

The *Profane form* came with programs that allow to build and explore geometric figures. The 1988 Turing prize was awarded to Ivan Sutherland for his pioneering work in the area of computer graphics. The program Sketchpad changed the way people interacted with computers, from non-graphical to graphical (Sutherland, 1963, 2003). While the original aim was to make computers more accessible, introducing graphical manipulations, while retaining the powers of abstraction that are critical to programmers, the direct manipulation interfaces have since succeeded by reducing the levels of abstraction exposed to the user.

The program Sketchpad can be considered as the point of origin for today's computer-aided graphic design programs (CAD). Not detracting from CAD programs, they are of little interest for the geometry practitioner, they are very high-precision tools to draw figures, e.g. for architects, drawing building plans, but they miss the step from drawings (static object) to figures (geometric construction), i.e. a set of objects and geometric relations between them (dynamic object). Meanwhile, dynamic geometry systems (DGS) allow building geometric constructions from free objects and elementary constructions. It became possible to manipulate the free objects (objects universally quantified), preserving the geometric properties of the construction.

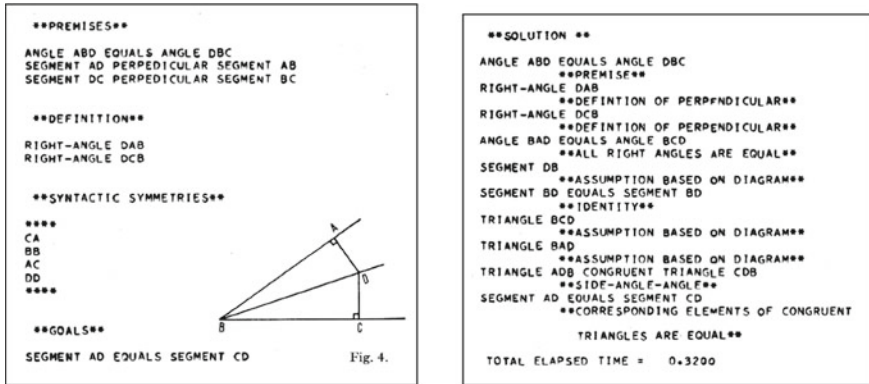


Fig. 2 Gelernter—Angle bisection

The first software packages that can be classified as dynamic geometry systems were Geometer's Sketchpad (Jackiw, 2001), which appeared first in 1989, and Cabri Géomètre (Laborde & Strässer, 1990), dating back to 1988, and they started another revolution: computers could be used in school for teaching geometry. Since then DGS become mature tools used by millions of users all over the world.

Dynamic geometry systems gave us the profane side of proofs in geometry. For example Fig. 2 (if done with a modern DGS) would have the points A , B and C as free points and point D as a constructed point (intersection of lines), moving the free points we can conjecture that the segments AD and CD are equal in length, i.e. we are exploring “all” possible configurations for a given geometric construction in the Cartesian model. Although those manipulations are not formal proofs because only a finite set of positions are considered and visualization can be misleading, they provide a first clue to the truthfulness of a given geometric conjecture.

The DGS and GATP are in a collision course and that is a good thing. From the development of GATP and DGS as completely separated tools, to the implementation of some GATP method in a DGS (e.g. *Cinderella*) or graphical components into a GATP (e.g. *GCLC* and *JGEX*) to the integration of GATP and DGS (e.g. *GeoGebra*). The fully integration of automated deduction components in other software is becoming a reality and it is expected that in a near future it will be possible to have those components broadly available.

Overview of the chapter In Sect. 2, the evolution of automated deduction in geometry is presented and in Sect. 3 the integration of GATP and DGS is discussed. In Sect. 4 other lines of research are presented and in Sect. 5 conclusions are drawn.

2 Automated Deduction in Geometry

For the last five decades, automated deduction in geometry, the sacred form, has been an important field in the area of automated reasoning. Various methods and techniques have been studied and developed for automatically proving and discovering geometric theorems (Chou, 1987; Chou & Gao, 2001; Chou et al., 1994).

2.1 Synthetic Methods

Adapting general-purpose reasoning approaches developed in the field of artificial intelligence (in the 60s of the twentieth century), synthetic methods, such as the approaches by Gelernter (1995, 1960, Nevis (1975), Elcock (1977), Greeno et al. (1979), Coelho and Pereira (1979, 1986), Zhang et al. (1995), were dedicated to automating traditional proving processes (Chou & Gao, 2001). Making use of axiomatic systems close to the ones used in secondary schools these systems tried to provide readable (by students and teachers) proofs. See Fig. 2 for an example of such proofs, from the GATP by Gelernter.

In many of these first attempts, the diagrams were used as a model (Coelho & Pereira, 1986):

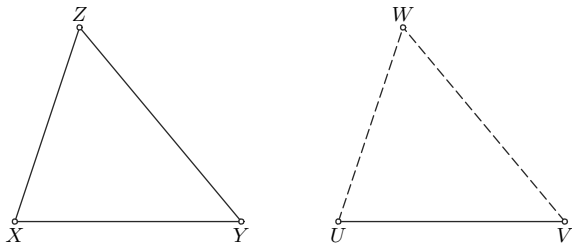
- the diagram as a filter (acting as a counter-example);
- the diagram as a guide (acting as an example, suggesting eventual conclusions).

As a filter the diagram permits to test the non-provability of a candidate sub-goal, pruning the proof tree.

As a guide the diagram can be used as a positive indication. Quoting from Coelho et al. (1986) Coelho and Pereira (1986) (see Fig. 3):

We want to prove two equal segments $UV = XY$, by congruent triangles. Suppose triangle XYZ exists, and our purpose is to find a triangle UVW on UV to compare to triangle XYZ . We need to search for existing or generated triangles on UV . The first thing is to find a convenient third point W , which must be different from U and V . The possible coordinates of the sought point W are computed from the coordinates of X, Y, Z, U and V , and a check is made in the diagram to see if a point with such coordinates exists. The diagram is used in a positive way for computing the possible coordinates for W .

Fig. 3 Coelho et al. (1986)—Diagram as a guide



The possibility of having geometric proofs, with natural language and (eventually) visual renderings, is a key aspect of this approach. Unfortunately, the combinatorial explosion while applying postulates implied the use of suitable heuristics that narrow the scope of the GATP and prevent the development of a general-purpose efficient GATP.

New synthetic approaches are being proposed. The geometric deductive database method combines the full-angle axiom system with the techniques of deductive databases, to develop an efficient GATP capable of proving a large set of geometric problems (Chou et al., 2000). A coherent logic² based GATP, *ArgoCLP*, is being developed which can be used to generate both readable and formal (machine verifiable) proofs in various theories, primarily geometry. The possibility of, using a top-down approach (from the conjecture to the conclusion), producing natural language proofs is a positive point, but, efficiency considerations are still a major concern (Stojanović et al., 2011).

2.2 Algebraic Methods

A different approach is given by the algebraic style methods, given by the translation of geometric problems to algebraic problems and the subsequent development of the proof by the application of algebraic manipulations. The characteristic set method, also known as Wu's method (Chou, 1985; Wu, 1984), the elimination method (Wang, 1995), the Gröbner bases method (Kapur, 1986a, b) and the Clifford algebra approach (Li, 2000) are examples of practical methods based on the algebraic approach.

Let us consider, for example, the Euler's Line theorem.

Theorem 1 (Euler's Line Theorem) *In any given triangle, the orthocentre, the centroid and the centre of the circumscribed circle are collinear (Fig. 4).*

Transcribing it to algebraic form we get (GATP: *JGEX*, Wu's method):

The Algebraic Form:

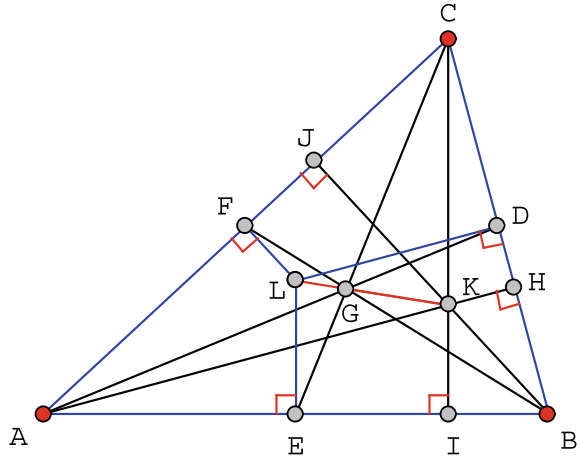
A: (0,0) B: (0,x4) C: (x5,x6) D: (x7,x8) E: (0,x10)
 F: (x11,x12) G: (x13,x14) H: (x15,x16) I: (0,x18)
 J: (x19,x20) K: (x21,x22) L: (x23,x24) M: (x25,x26)

The Equational Hypotheses:

1: D : midpoint(BC) $2x8 - x6 - x4 = 0$ $2x7 - x5 = 0$
 2: E : midpoint(BA) $2x10 - x4 = 0$
 3: F : midpoint(AC) $2x12 - x6 = 0$ $2x11 - x5 = 0$
 :
 11: LF \perp CA $x6x24 + x5x23 - x6x12 - x5x11 = 0$
 12: M : on line EL $x23x26 + (-x24 + x10)x25 - x10x23 = 0$

² Coherent logic is a fragment of (finitary) first-order logic which allows only the connectives and quantifiers \wedge (and), \vee (or), \top (true), \perp (false), \exists (existential quantifier).

Fig. 4 Euler’s line theorem, JGEX construction



At this stage the geometric/algebraic connection is still possible, each geometric element has a correspondent set of algebraic equations and vice versa. In the proof itself, pure algebraic methods are used. In Wu’s method, this implies calculating triangular systems and pseudo-remainders (Chou, 1985; Wu, 1984).

The JGEX’s proof (Wu’s method) is:

The Triangularized Hypotheses (TS):

$$h_0: 2x_7 - x_5 = 0$$

$$h_1: 2x_8 - x_6 - x_4 = 0$$

$$h_2: x_9 = 0$$

⋮

$$h_{18}: 2x_5x_{25} + (-2x_6 + 2x_4)x_8 + x_4x_6 - x_5^2 - x_4^2 = 0$$

$$h_{19}: (2x_6 - 2x_4)x_{26} + 2x_5x_{25} + (-2x_6 + 2x_4)x_8 - x_5^2 = 0$$

The Conclusion (CONC):

L, G, K are collinear

$$(x_{21} - x_{13})x_{24} + (-x_{22} + x_{14})x_{23} + x_{13}x_{22} - x_{14}x_{21} = 0$$

Successive Pseudo Remainder of CONC wrpt TS :

$$R_{18} = [x_{24}, 6]$$

$$R_{17} = \text{prem}(R_{18}, h_{17}) = [x_{23}, 6]$$

⋮

$$R_7 = \text{prem}(R_8, h_7) = [x_{13}, 6]$$

$$R_6 = \text{prem}(R_7, h_6) = [0, 0]$$

$$\text{Remainder} = R_6 = 0$$

The conclusion is true

GATP based on these methods are efficient and able to proof a large number of geometric conjectures. The price to pay is the absence of geometric proofs and the algebraic proofs, if any, are only (barely) readable by experts (Chou & Gao, 2001).

2.3 Semi-Synthetic Methods

In order to combine the readability of synthetic methods and the efficiency of algebraic methods, some approaches, such as the area method (Chou et al., 1996a; Janičić et al., 2012) and the full-angle method (Chou et al., 1996b), represent geometric knowledge in the form of expressions with respect to geometric invariants.

For stating and proving conjectures, these methods use a set of specific *geometric quantities* that enable treating geometric relations defining an axiom system. Considering the area method, we have:³

- *ratio of parallel directed segments*, denoted $\overline{AB}/\overline{CD}$. If the points A, B, C and D are collinear, $\overline{AB}/\overline{CD}$ is the ratio between the lengths of directed segments AB and CD . If the points A, B, C and D are not collinear, and it holds $AB \parallel CD$, there is a parallelogram $ABPQ$ such that P, Q, C and D are collinear and then $\frac{\overline{AB}}{\overline{CD}} = \frac{\overline{QP}}{\overline{CD}}$.
- *signed area* for a triangle ABC , denoted S_{ABC} is the area of the triangle ABC , multiplied by -1 , if ABC has the negative orientation.
- *Pythagoras difference*,⁴ denoted \mathcal{P}_{ABC} , for the points A, B, C , defined as $\mathcal{P}_{ABC} = \overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$.

An axiom system based on these three geometric quantities allows expressing (in form of equalities) geometry properties such as collinearity of three points ($S_{ABC} = 0$), parallelism of two lines ($S_{ABC} = S_{BCD}$), equality of two points ($\mathcal{P}_{ABA} = 0$), perpendicularity of two lines ($\mathcal{P}_{ACD} = \mathcal{P}_{BCD}$), etc. (Chou et al., 1996a; Janičić et al., 2012).

To prove a given conjecture, we have to express the hypotheses of a theorem using a set of starting (“free”) points and a set of constructive statements, each of them introducing a new point, and to express the conclusion by an equality between polynomials in the geometric quantities of the method (without considering Cartesian coordinates). The proof is developed by eliminating, in reverse order, the points introduced before, using for that purpose a set of appropriate lemmas (see Fig. 5). After eliminating all the introduced points, the goal equality of the conjecture collapses to an equality between two rational expressions involving only free points. This equation can be further simplified to involve only independent variables. If the expressions on the two sides are equal, the conjecture is a theorem, otherwise it is not.

In the example above (see Theorem 1), the conjecture can be expressed using the signed area of triangles, $S_{LGK} = 0$, i.e. the signed area of the triangle LGK is zero, so we are considering a degenerate triangle, so the points are collinear. Using the area method, both *GCLC* and *JGEX* failed to prove this conjecture (only using the

³ Negative and positive orientation are only a syntactic convention to disambiguate between “different” geometric constructions built from the same set of points.

⁴ 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. For a triangle ABC with the right angle at B , it holds that $\mathcal{P}_{ABC} = 0$.

$$\begin{aligned}
 (1) \quad & \frac{\overrightarrow{HT}}{\overrightarrow{TO}} = 2, \text{ by the statement} \\
 (2) \quad & \left(-1 \cdot \frac{\overrightarrow{TH}}{\overrightarrow{TO}}\right) = 2, \text{ by geometric simplifications} \\
 (3) \quad & \left(-1 \cdot \frac{S_{TAA_2}}{S_{TAOA_2}}\right) = 2, \text{ by Lemma 37, first case — assuming points } T, A, \text{ and } A_2 \text{ are not collinear (point } H \text{ eliminated)} \\
 (4) \quad & \frac{(-1 \cdot S_{TAA_2})}{S_{TAOA_2}} = 2, \text{ by algebraic simplifications} \\
 (5) \quad & \frac{\left(-1 \cdot \frac{((P_{ABC} \cdot S_{TAC}) + (P_{ACB} \cdot S_{TAB}))}{P_{BCB}}\right)}{S_{TAOA_2}} = 2, \text{ by Lemma 31 (point } A_2 \text{ eliminated)} \\
 (6) \quad & \frac{((-1 \cdot (P_{ABC} \cdot S_{TAC})) + (-1 \cdot (P_{ACB} \cdot S_{TAB})))}{(P_{BCB} \cdot S_{TAOA_2})} = 2, \text{ by algebraic simplifications} \\
 (7) \quad & \frac{((-1 \cdot (P_{ABC} \cdot S_{TAC})) + (-1 \cdot (P_{ACB} \cdot S_{TAB})))}{\left(P_{BCB} \cdot \frac{((P_{ABC} \cdot S_{TAOC}) + (P_{ACB} \cdot S_{TAOB}))}{P_{BCB}}\right)} = 2, \text{ by Lemma 31 (point } A_2 \text{ eliminated)} \\
 (8) \quad & \frac{((-1 \cdot (P_{ABC} \cdot S_{TAC})) + (-1 \cdot (P_{ACB} \cdot S_{TAB})))}{((P_{ABC} \cdot S_{TAOC}) + (P_{ACB} \cdot S_{TAOB}))} = 2, \text{ by algebraic simplifications} \\
 (9) \quad & \frac{((-1 \cdot (P_{ABC} \cdot S_{TAC})) + (-1 \cdot (P_{ACB} \cdot S_{TAB})))}{((P_{ABC} \cdot (S_{TAO} + S_{CTO})) + (P_{ACB} \cdot (S_{TAO} + S_{BTO})))} = 2, \text{ by geometric simplifications} \\
 & \vdots \\
 (1082) \quad & 0 = 0, \text{ by algebraic simplifications}
 \end{aligned}$$

NDG conditions are:

- $S_{ACC_1} \neq S_{A_1CC_1}$ i.e., lines AA_1 and CC_1 are not parallel (construction based assumption)
- $S_{M_a^0M_a^2T_a^3} \neq S_{T_a^1M_a^2T_a^3}$ i.e., lines $M_a^0T_a^1$ and $M_a^2T_a^3$ are not parallel (construction based assumption)
- $S_{ACC_2} \neq S_{A_2CC_2}$ i.e., lines AA_2 and CC_2 are not parallel (construction based assumption)
- $P_{TOT} \neq 0$ i.e., points T and O are not identical (conjecture based assumption)
- $S_{ACC_1} \neq 0$ i.e., points A, C and C_1 are not collinear (cancellation assumption)
- $S_{ACB} \neq 0$ i.e., points A, C and B are not collinear (cancellation assumption)
- $S_{TAA_2} \neq 0$ i.e., points T, A and A_2 are not collinear (made for Lemma 37)

Number of elimination proof steps: 386; Number of geometric proof steps: 876;
 Number of algebraic proof steps: 5982; Total number of proof steps: 7244;
 Time spent by the prover: 4.902 seconds; Number of pages: 169

Fig. 5 Euler’s line theorem, *GCLC* proof (area method)

algebraic methods they were capable of proving it), but for the same construction the conjecture $\overline{GK}/\overline{LG} = 2$ is provable. The proof itself is a (readable) sequence of steps performing algebraic simplifications, geometric simplifications and the application of lemmas allowing to eliminate the free points, but with 169 pages this formal proof cannot be considered readable (see Fig. 5).

The GATP implementing these methods are efficient and capable of producing formal, but readable (in some cases, small) proofs. Given the fact that these methods

do not use the usual axioms systems used in secondary schools, the proofs produced are readable, even by secondary schools' students and teachers, but not without a previous study of the axiom system used by the GATP.

2.4 Generic First-Order Provers

First-order ATP must also be considered. Provers like *Vampire*, *Leo* and many others⁵ are very efficient⁶ and capable of proving theorems in many areas. Using axioms systems for geometry, e.g. Tarski, they can be used in geometry also. For example, the *Thousands of Problems for Theorem Provers* (TPTP) repository (Sutcliffe, 2017) has the *GEO* domain with many problems specified in the first-order logic (FOL), using the Tarski Geometry (Quaife, 1989) or the geometric deductive database method (GDDM) (Chou et al., 2000)⁷ axiom systems. The proof of those conjectures can then be attempted by the different ATP.

For example, the Euler's line theorem can be expressed in the following form (using the GDDM):

```
%----Include Geometry Deductive Database Method axioms
include('geometryDeductiveDatabaseMethod.ax').

fof(exemploeulerLineGEO0315a,conjecture,(
  ! [ A,B,C,D,E,F,G,H,I,J,K,L,M ] :
  ((midp(D,B,C) & midp(E,B,A) & midp(F,A,C) & coll(G,B,F)
    & coll(G,C,E) & perp(H,A,B,C) & coll(H,B,C)
    & perp(I,C,A,B) & coll(I,A,B) & perp(J,B,A,C)
    & coll(J,A,C) & coll(K,A,H) & coll(K,C,I) & perp(B,A,M,E)
    & perp(C,B,L,D) & perp(C,A,L,F) & coll(M,D,L))
  =>
  (coll(L,G,K))) ).
```

the specification of the problem is done in *First-Order Form* (FOF). Using the ATP *Vampire*, the conjecture is proved by refutation; below an excerpt of the proof is given:

```
% Refutation found. Thanks to Tanya!
% SZS status Theorem for eulerLineGEO0315a
% SZS output start Proof for eulerLineGEO0315a
1. ! [X0,X1,X2] : (coll(X0,X1,X2) => coll(X0,X2,X1)) [input]
2. ! [X0,X1,X2] : (coll(X0,X1,X2) => coll(X1,X0,X2)) [input]
   :
2754. $false <- (74) [resolution 1962,1558]
```

⁵ https://en.wikipedia.org/wiki/Automated_theorem_proving.

⁶ <http://www.tptp.org/CASC/>.

⁷ This is a recent contribution, by the author, to TPTP.

```

2765. ~74 [avatar contradiction clause 2754]
2766. $false [avatar sat refutation 1963,2749,2765]
% SZS output end Proof for eulerLineGEO0315a
% -----
% Version: Vampire 4.5.1
% Termination reason: Refutation

% Memory used [KB]: 6652
% Time elapsed: 0.062 s

```

Using this approach, the proof script (if any) would be a logical proof (e.g. using the resolution method). As can be seen above, in the case of the proof done by the ATP *Vampire*, the proof contains information about the lemmas used and, if some post-processing is used, can be readable, but the connection with the geometric construction is difficult, at best.

An advantage of this approach is the availability of many generic ATPs with very efficient implementations.⁸ The unavailability of geometric proofs is the drawback.

2.5 Other Approaches

Rule-based approaches explore the possibility of building a sound, not necessarily complete, axiom system. The idea is to have a minimal set of axioms, lemmas and rules of inference that can characterize a given sub-area of geometry (Pambuccian, 2004).

One example of such an approach is given by the tutorial system *QED-tutrix*. Exploring the logic programming language *Prolog*⁹ (Clocksin & Mellish, 2003) a set of axioms, lemmas and rules of inference, adapted to the type of problems at hand, are implemented and explored by the *Prolog* rule-based logical query mechanism. The *QED-tutrix* tutorial system builds the *Hypothesis, Properties, Definitions, Intermediate results and Conclusion graph* (HPDIC-graph). The HPDIC-graph contains all possible proofs for a given problem, using a given set of axioms. Having that (possibly, very large) graph the system can help the learner, validating the steps already taken and providing hints for the next steps (Font et al., 2018; Gagnon et al., 2017).

A project, still in its early stages, uses *Maude*¹⁰ (Clavel et al., 2007), an equational (and rewriting) logic system to implement the Tarski axiom system as described by Art Quaife (1989) (see Fig. 6).

⁸ For the Euler line theorem, *GCLC*'s Wu Method took 0.012s, *Vampire* took 0.062s.

⁹ <https://en.wikipedia.org/wiki/Prolog>.

¹⁰ http://maude.cs.illinois.edu/w/index.php/The_Maude_System.

```

*** System Tarski over G3cp
***
fmod FORMULA is
pr QID . *** Maude's Qualified Identifiers ('a','b', etc).
sort Prop . *** Atomic propositions
sort Formula . *** Formulas
sort Point . *** Points
sort Segment . *** Segments
subsort Qid < Prop < Formula .
subsort Qid < Point .
(...)
*** Tarski geometry primitive relations
op p : -> Point [ctor] .
op p1 : -> Point [ctor] .
op p11 : -> Point [ctor] .
op _*_ : Point Point -> Segment [ctor comm] .
op betweenness : Point Point Point -> Prop [ctor comm] .
op equidistance : Segment Segment -> Prop [ctor comm] .
op extension : Point Point Point Point -> Point . *** extension
op innerPasch : Point Point Point Point Point -> Point .
op euclides1 : Point Point Point Point Point -> Point .
op euclides2 : Point Point Point Point Point -> Point .
op continuity : Point Point Point Point Point Point -> Point .
op _==_ : Point Point -> Formula [ctor comm] .
endfm
mod Tarski is
***
*** Tarski' Geometry (Art Quaife (1989), JAR 5, 97--118.
(...)
*** A7 Inner Pasch
rl [ip1] : C , betweenness(U,V,W) , betweenness(Y,X,W) |-- betweenness(V,innerPasch(U,V,W,X,Y),Y) , C' =>
  proved .
rl [ip2] : C , betweenness(U,V,W) , betweenness(Y,X,W) |-- betweenness(X,innerPasch(U,V,W,X,Y),U) , C' =>
  proved .
(...)
*** A10 Euclid's axiom
rl [euc11] : C , betweenness(U,W,Y) , betweenness(V,W,X) |-- U == W , betweenness(U,V,euclides1(U,V,W,X,Y)) ,
  C' => proved .
rl [euc12] : C , betweenness(U,W,Y) , betweenness(V,W,X) |-- U == W , betweenness(U,X,euclides2(U,V,W,X,Y)) ,
  C' => proved .
rl [euc13] : C , betweenness(U,W,Y) , betweenness(V,W,X) |-- U == W , betweenness(euclides1(U,V,W,X,Y),
  euclides2(U,V,W,X,Y)) , C' => proved .
(...)

```

Fig. 6 Implementation, as a Maude module, of Tarski axiom system, as described by Art Quaife

Both approaches share the use of logic programming languages where the introduction of specific lemmas, for specific purposes, can be easily made, e.g. SSS, SAS, ASA and AAS lemmas for the congruence of triangles, or the alternate interior and exterior angles of parallel lines.

As stated above, this systems can be used to implement sound, but not necessarily complete, axiom systems. These kind of system can be useful in specific situations, e.g. in secondary schools mathematics classes.

3 Dynamic Geometry

Dynamic Geometry Systems

Dynamic geometry can be characterized by the construction of geometric figures (as opposed of fixed drawings), built from free objects, i.e. universally quantified objects (e.g. points) and elementary properties preserving constructions.

Analysing the example given in Fig. 7, the construction has A, B, D and E as free points, these can be moved freely in the plane, points C and F belongs to AB and DE , respectively, they have only one degree of freedom, being able to move in the line they belong and, finally, points G, H and I are the intersections of two lines, they do not have any degree of freedom.

Dynamic geometry systems give us the profane side of proofs in the Cartesian model of Euclidean geometry. By moving the free points, we can conjecture that the points G, H and I are collinear (red line), i.e. we are exploring “all” possible configurations for that geometric construction. Although these manipulations are not formal proofs because only a finite set of positions are considered and because visualisation can be misleading, they provide a first clue to the truthfulness of a given geometric conjecture.

Dynamic geometry systems are now mature software tools with a very large users base. The dynamic character of these programs give to its users the possibility of building dynamic geometric constructions, exploring conjectures about them, so

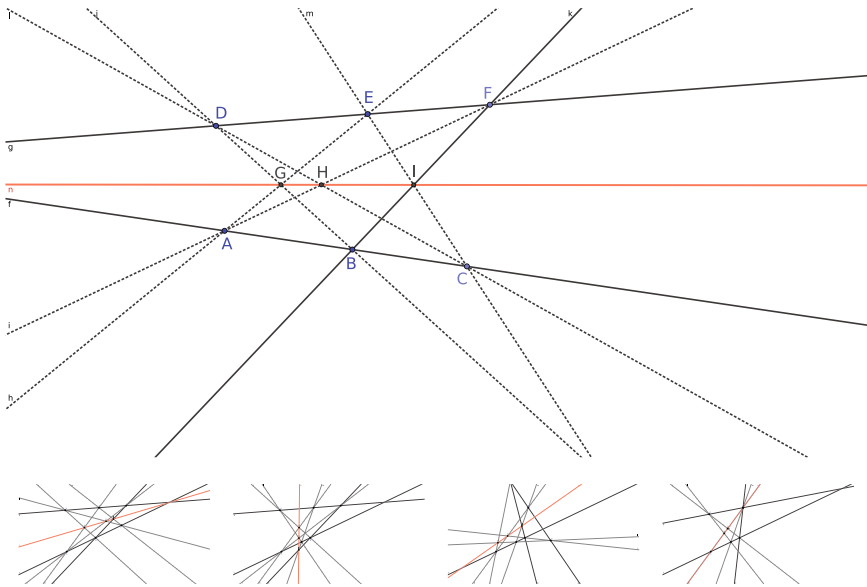


Fig. 7 Pappus' Hexagon Theorem, DGS construction

they give a first, informal, opportunity to explore geometric proofs. But to avoid the visual proofs pitfall, the connection with geometric automated theorem provers (GATP) should be considered.

Dynamic Geometry Systems and Geometry Automated Theorem Provers

There are already some systems combining DGS and GATP. The following list is organized as follows: from GATP and GATP with rendering capabilities to DGS with some automatic proving capabilities built-in, ending with environments with DGS and/or GATP capabilities.

Open Geometry Prover *The Open Geometry Prover* (OGP) is an open source project,¹¹ aiming to implement various geometry automated theorem provers.

It can be used as a stand-alone tool but can also be integrated into other geometry tools, such as dynamic geometry software. In its current state, OGP implements the Wu's algebraic methods. Some partial work has been made in the semi-synthetic methods: the area method and the full-angle method (Baeta & Quaresma, 2013; Botana et al., 2015; Petrović et al., 2012).

GCLC *Geometry Constructions → LaTeX Converter (GCLC)*, an open source GATP with a graphics engine, for the Wu, Gröbner bases and area methods. It is possible to add a conjecture to a given geometric construction (with a graphical rendering) and ask for its proof with natural language rendering (Janičić, 2006).

JGEX *Java Geometry Expert* (JGEX), a GATP with graphics engine, for the Wu, Gröbner bases, area, full-angle and deductive database methods. It is possible to add a conjecture to a given geometric construction and ask for its proof with natural and visual language renderings (Ye et al., 2011).

Cinderella *The Interactive Geometry Software Cinderella* is a DGS with a randomized theorem checker (Kortenkamp & Richter-Gebert, 2004; Richter-Gebert & Kortenkamp, 1999).

The last one has a different approach from the first three, it is not a prover capable of a formal proof, but more a model checker, capable of generating many random instances for a given geometric configuration. So, for a given conjecture, we obtain a probabilistic answer to its validity (Kortenkamp & Richter-Gebert, 2004).

Apart from the library OGP, these systems¹² are monolith systems, i.e. even if the GATP are modules that can be used by themselves (e.g. the *GCLC*), the DGS and the GATP are tightly integrated. The DGS are not able to use external GATP and the GATP are not prepared to be integrated in other DGS.

A modular approach is beginning to make its way, i.e. an approach where DGS and GATP can be developed by different teams and nevertheless be combined. Examples of such systems are:

¹¹ The OpenGeometryProver github project: <https://github.com/opengeometryprover/>.

¹² JGEX is currently not being supported/developed.

GeoProof A DGS that interfaces with the *Coq* Prover Assistant (Bertot & Castéran, 2004), allowing to check proofs built interactively (Narboux, 2007).

GeoGebra The DGS embedded prover system chooses one of the available methods and translates the problem specified by the end user as the input for the selected method, similarly to portfolio solvers.¹³ The available GATP implement the Wu's method, the Buchberger–Kapur method, the area method and Recio's exact check method. The separation between the GATP and the DGS opens the possibility of using third-party GATP. For example, the *Open Geometry Prover* Wu's method GATP. From the DGS interface the user can ask for the validation of a given geometric conjecture, related to the construction, e.g. `Prove AreCollinear(G, H, I)` is the command that a user may use to prove, formally, that the points G , H and I in the construction of Fig. 7 are collinear. For the moment, this is only a formal validation, no proof script is available (Botana et al., 2015; Kovács, 2015; Kovács & Recio, 2020; Nikolić et al., 2019).

This project can be visited at <https://www.geogebra.org/m/McEqwQNb>.

When applied to areas such as education GATP and DGS are being combined in environments that use both tools for the purpose of learning.

QED-Tutrix *QED-Tutrix* is an intelligent tutor for geometry (see Sect. 2.5) offering an interface to help high school students to freely explore the geometric problems and their proofs.

For each geometric conjecture, the system builds the tree of all possible proofs, allowing the students to try to prove the conjecture, with the system helping in each step of the proof (Font et al., 2018; Gagnon et al., 2017).

Geometriagon *Geometriagon*¹⁴ is a project to explore geometric constructions made with ruler and compass. The dynamic geometry software supporting the project is C.a.R.¹⁵ It provides an extensive list of problems to be solved by registered users, validating (or not) the solution found.

A similar approach, not in the area of geometry, is given by *Edukera*,¹⁶ a Web-environment to teach Logic and Mathematics (calculus and sets), with the assistance of the *Coq* proof assistant.

¹³ Portfolio problem solving is an approach in which for an individual instance of a specific problem, one particular, hopefully most appropriate, solving technique is automatically selected among several available ones and used. The selection usually employs machine learning methods.

¹⁴ <http://polarprof-001-site1.htmpurl.com/geometriagon/>.

¹⁵ http://car.rene-grothmann.de/doc_en/index.html.

¹⁶ <https://www.edukera.com/>.

4 Other Lines of Research

The usefulness of automated deduction methods and tools in geometry does not circumscribe itself to its use in dynamic geometry systems:

Repositories of Geometric Knowledge In the repository *Thousand of Geometric problems for geometric Theorem Provers* (TGTP),¹⁷ a list of more than two hundred geometric conjectures and their proofs by several GATP are kept (Quaresma, 2011). It implements a protocol for the exchange of geometric information between applications, allowing a direct access to its repository from external tools.

In the repository *Thousands of Problems for Theorem Provers* (TPTP) (Sutcliffe, 2017),¹⁸ a section is dedicated to geometric problems (in this repository the problems are specified as first order logic problems).

Recent efforts are being made to start a *Geometry Automated Provers Competition* (Baeta et al., 2020), in order to help improve the efficiency and usefulness of GATP.

Knowledge Management Besides the efforts to implement readable formal proofs produced by GATP, there are other avenues of research that are being explored. For example: to evaluate the complexity and also the interestingness of geometric proofs (Gao et al., 2019; Quaresma et al., 2020);¹⁹ to explore the automatic discovery of theorems in elementary geometry (Recio & Vélez, 1999); to implement a semantic geometric search mechanism, i.e. the possibility of, having a given geometric construction, to search for other congruent geometric constructions or even other geometric constructions with some common geometric properties. A prototype of this search mechanism is already implemented in the *TGTP* repository (Haralambous & Quaresma, 2014, 2018).

All these efforts are being made to help answer the questions related to the use of GATP as one more tool in the geometer's toolbox.

5 Conclusions

It is clear that automated deduction methods and tools are beginning to make their way into many different uses, in many different contexts. However, there are many issues to be solved before a more complete integration of automated deductive tools can take place. Examples of problems related with the use of automated deduction in geometry tools are: application programming interface; common formats for information interchange; natural and visual languages renderings of the proofs; repositories of

¹⁷ <http://hilbert.mat.uc.pt/TGTP/index.php>.

¹⁸ <http://www.tptp.org/>.

¹⁹ Pedro Quaresma and Pierluigi Graziani, Measuring the Readability of a Proof, submitted to publication.

geometric knowledge with powerful search mechanisms; proof discovery; problems and proof classification for complexity and efficiency.

The complexity and the sheer size of the task led to recent efforts to establish a network of researchers working in the area of formal reasoning, knowledge-based intelligent software and geometric knowledge management. The network will need to focus in the creation of an intelligent computational environment in which advanced software tools and deduction mechanisms are embedded for symbolic-numeric geometric computation, interactive or automated geometric reasoning, knowledge validation, knowledge discovery and knowledge management. Such a “superset of a book” of geometric knowledge with embedded tools, freely available in all computational platforms, adaptable, collaborative and adaptive to each and every user’s profiles, would bring together a whole new generation of mathematical tools with impact at all levels: exploratory research, applications in mathematics and education.

All these efforts are leading to an integration of formal deduction and dynamic geometry in vivid environments, to be used in the exploration of mathematics in its fullness. It is a complex task, but it is also an exciting task, as new advances in many areas are expected in the near future.

References

- Baeta, N., & Quaresma, P. (2013). The full angle method on the OpenGeoProver. In C. Lange, D. Aspinall, J. Carette, J. Davenport, A. Kohlhase, M. Kohlhase, P. Libbrecht, P. Quaresma, F. Rabe, P. Sojka, I. Whiteside, & W. Windsteiger (Eds.), *MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at the Conference on Intelligent Computer Mathematics, no. 1010 in CEUR Workshop Proceedings*. Aachen. <http://ceur-ws.org/Vol-1010/paper-08.pdf>.
- Baeta, N., Quaresma, P., & Kovács, Z. (2020). Towards a geometry automated provers competition. In *Proceedings 8th International Workshop on Theorem Proving Components for Educational Software, (ThEdu'19)*. Electronic Proceedings in Theoretical Computer Science (Vol. 313, pp. 93–100). Natal, Brazil, 25th August 2019. <https://doi.org/10.4204/EPTCS.313.6>
- Beeson, M. (2015). A constructive version of Tarski’s geometry. *Annals of Pure and Applied Logic*, 166(11), 1199–1273. <https://doi.org/10.1016/j.apal.2015.07.006>.
- Bertot, Y., & Castéran, P. (2004). *Interactive theorem proving and program development: Coq’Art: The calculus of inductive constructions*. EATCS: Springer. <https://doi.org/10.1007/978-3-662-07964-5>.
- Botana, F., Hohenwarter, M., Janičić, P., Kovács, Z., Petrović, I., Recio, T., & Weitzhofer, S. (2015). Automated theorem proving in GeoGebra: Current achievements. *Journal of Automated Reasoning*, 55(1), 39–59. <https://doi.org/10.1007/s10817-015-9326-4>.
- Chou, S. (1985). *Proving and discovering geometry theorems using Wu’s method*. Ph.D. thesis, The University of Texas, Austin.
- Chou, S. C. (1987). *Mechanical geometry theorem proving*. Dordrecht: D. Reidel Publishing Company.
- Chou, S. C., & Gao, X. S. (2001). Automated reasoning in geometry. In J. A. Robinson & A. Voronkov (Eds.), *Handbook of automated reasoning* (pp. 707–749). Elsevier Science Publishers B.V. <https://doi.org/10.1016/B978-044450813-3/50013-8>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1993). Automated production of traditional proofs for constructive geometry theorems. In M. Vardi (Ed.), *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science LICS* (pp. 48–56). IEEE Computer Society Press.

- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1994). *Machine proofs in geometry*. World Scientific. <https://doi.org/10.1142/2196>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1995). Automated production of traditional proofs in solid geometry. *Journal of Automated Reasoning*, 14, 257–291. <https://doi.org/10.1007/bf00881858>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1996a). Automated generation of readable proofs with geometric invariants, I. Multiple and shortest proof generation. *Journal of Automated Reasoning*, 17(13), 325–347. <https://doi.org/10.1007/BF00283133>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (1996b). Automated generation of readable proofs with geometric invariants, II. Theorem proving with full-angles. *Journal of Automated Reasoning*, 17(13), 349–370. <https://doi.org/10.1007/BF00283134>.
- Chou, S. C., Gao, X. S., & Zhang, J. Z. (2000). A deductive database approach to automated geometry theorem proving and discovering. *Journal of Automated Reasoning*, 25, 219–246.
- Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., & Talcott, C. (2007). *All about Maude: A high-performance logical framework*. Lecture Notes in Computer Science (Vol. 4350). Springer. <https://doi.org/10.1007/978-3-540-71999-1>.
- Clocksin, W. F., & Mellish, C. S. (2003). *Programming in Prolog*. Springer, Berlin, Heidelberg. <https://doi.org/10.1007/978-3-642-55481-0>.
- Coelho, H., & Pereira, L. M. (1979). Geom: A Prolog geometry theorem prover. Memórias 525, Laboratório Nacional de Engenharia Civil, Ministério de Habitação e Obras Públicas, Portugal.
- Coelho, H., & Pereira, L. M. (1986). Automated reasoning in geometry theorem proving with Prolog. *Journal of Automated Reasoning*, 2(4), 329–390. <https://doi.org/10.1007/BF00248249>.
- Collins, G. (1975). Quantifier elimination for real closed fields by cylindrical algebraic decomposition. In H. Brakhage (Ed.), *Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern*. Lecture Notes in Computer Science (Vol. 33), 20–23 May 1975. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-07407-4_17.
- van Dalen, D. (1980). *Logic and structure*. Universitext. Springer-Verlag
- Elcock, E. W. (1977). Representation of knowledge in geometry machine. *Machine Intelligence*, 8, 11–29.
- Font, L., Richard, P. R., & Gagnon, M. (2018). Improving QED-Tutrix by automating the generation of proofs. In P. Quaresma, & W. Neuper (Eds.), *Proceedings 6th International Workshop on Theorem Proving Components for Educational Software*, Gothenburg, Sweden, 6 Aug 2017, Electronic Proceedings in Theoretical Computer Science (Vol. 267, pp. 38–58). Open Publishing Association (2018). <https://doi.org/10.4204/EPTCS.267.3>.
- Gabriel Silva, J. (2017). Coimbra: uma universidade global, desde o século xvi. *Rua Larga*, 50. Imprensa da Universidade de Coimbra
- Gagnon, M., Leduc, N., Richard, P., & Tessier-Baillargeon, M. (2017). Qed-tutrix: Creating and expanding a problem database towards personalized problem itineraries for proof learning in geometry. In *Proceedings of the Tenth Congress of the European Society for Research in Mathematics Education (CERME10)*.
- Gao, H., Li, J., & Cheng, J. (2019). Measuring interestingness of theorems in automated theorem finding by forward reasoning based on strong relevant logic. In *2019 IEEE International Conference on Energy Internet (ICEI)* (pp. 356–361). IEEE. <https://doi.org/10.1109/ICEI.2019.00069>.
- Gelernter, H. (1995). Realization of a geometry-theorem proving machine. In *Computers & thought*, 2nd edn. (pp. 134–152). MIT Press, Cambridge, MA, USA.
- Gelernter, H., Hansen, J. R., & Loveland, D. W. (1960). *Empirical explorations of the geometry theorem machine*. In Papers presented at the 3–5 May 1960, western joint IRE-AIEE-ACM computer conference, IRE-AIEE-ACM'60 (Western) (pp. 143–149). ACM, New York, NY, USA. <https://doi.org/10.1145/1460361.1460381>.
- Greeno, J., Magone, M. E., & Chaiklin, S. (1979). Theory of constructions and set in problem solving. *Memory and Cognition*, 7(6), 445–461. <https://doi.org/10.3758/BF03198261>.

- Haralambous, Y., & Quaresma, P. (2014). Querying geometric figures using a controlled language, ontological graphs and dependency lattices. In S. W. et al. (Eds.), *CICM 2014*. LNAI (Vol. 8543, pp. 298–311). Springer.
- Haralambous, Y., & Quaresma, P. (2018). Geometric search in TGTP. In H. Li (Ed.), *Proceedings of the 12th International Conference on Automated Deduction in Geometry*. SMS International. <http://adg2018.cc4cm.org/ADG2018Proceedings>.
- Hilbert, D. (1977). *Foundations of geometry*, 10th Revised edition. Paul Barnays: Open Court Publishing.
- Jackiw, N. (2001). *The Geometer's Sketchpad v4.0*. Key Curriculum Press.
- Janičić, P. (2006). GCLC—A tool for constructive Euclidean geometry and more than that. In A. Iglesias & N. Takayama (Eds.), *Mathematical Software, ICMS 2006*. Lecture Notes in Computer Science (Vol. 4151, pp. 58–73). Springer. https://doi.org/10.1007/11832225_6.
- Janičić, P., Narboux, J., & Quaresma, P. (2012). The area method: A recapitulation. *Journal of Automated Reasoning*, 48(4), 489–532. <https://doi.org/10.1007/s10817-010-9209-7>.
- Kapur, D. (1986a). Geometry theorem proving using Hilbert's Nullstellensatz. In *SYMSAC'86: Proceedings of the Fifth ACM Symposium on Symbolic and Algebraic Computation* (pp. 202–208). New York, NY, USA: ACM Press. <https://doi.org/10.1145/32439.32479>.
- Kapur, D. (1986b). Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4), 399–408. [https://doi.org/10.1016/S0747-7171\(86\)80007-4](https://doi.org/10.1016/S0747-7171(86)80007-4).
- Kortenkamp, U., & Richter-Gebert, J. (2004). Using automatic theorem proving to improve the usability of geometry software. In P. Libbrecht (Ed.), *Proceedings of MathUI 2004*. <http://kortenkamps.net/papers/2004/ATP-UI-article.pdf>.
- Kovács, Z. (2015). The relation tool in GeoGebra 5. In F. Botana & P. Quaresma (Eds.), *Automated deduction in geometry* (pp. 53–71). Springer International Publishing. https://doi.org/10.1007/978-3-319-21362-0_4.
- Kovács, Z., & Recio, T. (2020). Geogebra reasoning tools for humans and for automatons. In *Proceedings of the 25th Asian Technology Conference in Mathematics* (pp. 16–30). Mathematics and Technology, LLC. <https://doi.org/10.13140/RG.2.2.26851.58407>.
- Laborde, J. M., & Strässer, R. (1990). Cabri-géomètre: A microworld of geometry guided discovery learning. *International Reviews on Mathematical Education- Zentralblatt fuer Didaktik der Mathematik*, 90(5), 171–177.
- Li, H. (2000). Clifford algebra approaches to mechanical geometry theorem proving. In X. S. Gao & D. Wang (Eds.), *Mathematics mechanization and applications* (pp. 205–299). San Diego, CA: Academic Press. <https://doi.org/10.1016/B978-012734760-8/50009-0>.
- Nikolić, M., Marinković, V., Kovács, Z., & Janičić, P. (2019). Portfolio theorem proving and prover runtime prediction for geometry. *Annals of Mathematics and Artificial Intelligence*, 85, 119–146. <https://doi.org/10.1007/s10472-018-9598-6>.
- Narboux, J. (2007). A graphical user interface for formal proofs in geometry. *Journal of Automated Reasoning*, 39, 161–180. <https://doi.org/10.1007/s10817-007-9071-4>.
- Nevis, A. (1975). Plane geometry theorem proving using forward chaining. *Artificial Intelligence*, 6(1), 1–23. <http://hdl.handle.net/1721.1/6218>.
- Pambuccian, V. (2004). The simplest axiom system for plane hyperbolic geometry. *Studia Logica*, 77(3), 385–411. <https://doi.org/10.1023/B:STUD.0000039031.11852.66>.
- Petrović, I., Kovács, Z., Weitzhofer, S., Hohenwarter, M., & Janičić, P. (2012). Extending GeoGebra with automated theorem proving by using OpenGeoProver. In *Proceedings CADGME 2012*, Novi Sad, Serbia.
- von Plato, J. (1995). The axioms of constructive geometry. In *Annals of pure and applied logic* (Vol. 76, pp. 169–200). [https://doi.org/10.1016/0168-0072\(95\)00005-2](https://doi.org/10.1016/0168-0072(95)00005-2).
- Quaife, A. (1989). Automated development of Tarski's geometry. *Journal of Automated Reasoning*, 5, 97–118. <https://doi.org/10.1007/BF00245024>.
- Quaresma, P. (2011). Thousands of geometric problems for geometric theorem provers (TGTP). In P. Schreck, J. Narboux & J. Richter-Gebert (Eds.), *Automated deduction in geometry*. Lecture

- Notes in Computer Science (Vol. 6877, pp. 169–181). Springer. https://doi.org/10.1007/978-3-642-25070-5_10.
- Quaresma, P., Santos, V., Graziani, P., & Baeta, N. (2020). Taxonomy of geometric problems. *Journal of Symbolic Computation*, 97, 31–55. <https://doi.org/10.1016/j.jsc.2018.12.004>.
- Recio, T., & Vélez, M. P. (1999). Automatic discovery of theorems in elementary geometry. *Journal of Automated Reasoning*, 23, 63–82. <https://doi.org/10.1023/A:1006135322108>.
- Richter-Gebert, J., & Kortenkamp, U. (1999). *The interactive geometry software Cinderella*. Springer.
- Simões, C. (2007). Azulejos que ensinam: Entrevista a António Leal Duarte. *Gazeta de Matemática*, 153, 4.
- Stojanović, S., Pavlović, V., & Janičić, P. (2011). A coherent logic based geometry theorem prover capable of producing formal and readable proofs. In P. Schreck, J. Narboux & J. Richter-Gebert (Eds.), *Automated deduction in geometry*. Lecture Notes in Computer Science (Vol. 6877, pp. 201–220). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-25070-5_12.
- Sutcliffe, G. (2017). The TPTP problem library and associated infrastructure. From CNF to TH0, TPTP v6.4.0. *Journal of Automated Reasoning*, 59(4), 483–502. <https://doi.org/10.1007/s10817-017-9407-7>.
- Sutherland, I. E. (1963). *Sketchpad, a man-machine graphical communication system*. Ph.D. thesis, Massachusetts Institute of Technology, Lincoln Laboratory.
- Sutherland, I. E. (2003). *Sketchpad: A man-machine graphical communication system*. Technical Report. UCAM-CL-TR-574, University of Cambridge, Computer Laboratory. <http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-574.pdf>.
- Tarski, A. (1951). *A decision method for elementary algebra and geometry*. Technical Report. RAND Corporation.
- Tessier-Baillargeon, M., Leduc, N., Richard, P., & Gagnon, M. (2017). Étude comparative de systèmes tutoriels pour l'exercice de la démonstration en géométrie. *Annales de Didactique et de Sciences Cognitives*, 22, 91–117.
- Wang, D. (1995). Reasoning about geometric problems using an elimination method. In J. Pfalzgraf & D. Wang (Eds.), *Automated practical reasoning* (pp. 147–185). New York: Springer. https://doi.org/10.1007/978-3-7091-6604-8_8.
- Wu, W. T. (1984). On the decision problem and the mechanization of theorem proving in elementary geometry. In *Automated theorem proving: After 25 years*. Contemporary Mathematics (Vol. 29, pp. 213–234). American Mathematical Society.
- Ye, Z., Chou, S. C., & Gao, X. S. (2011). An introduction to java geometry expert. In T. Sturm & C. Zengler (Eds.), *Automated deduction in geometry*. Lecture Notes in Computer Science (Vol. 6301, pp. 189–195). Berlin, Heidelberg: Springer. https://doi.org/10.1007/978-3-642-21046-4_10.
- Zhang, J. Z., Chou, S. C., & Gao, X. S. (1995). Automated production of traditional proofs for theorems in Euclidean geometry I. The Hilbert intersection point theorems. *Annals of Mathematics and Artificial Intelligence*, 13, 109–137. <https://doi.org/10.1007/BF01531326>.