Geometric Automated Theorem Proving

Pedro Quaresma

CISUC, Mathematics Department
Faculty of Science and Technology, University of Coimbra

University of Urbino, January 2019

Geometric Automated Theorem Proving (GATP)

GATPs—Two major lines of research [CGZ94, CG01, Wan96]:

- Synthetic methods; Seminar 1
- Algebraic methods. Seminar 2

Formalization & Automated Discovery: Seminar 3

- Formalisation;
- Automated Discovery.

Geometric Tools & Geometric Knowledge Management: Seminar 4

- Geometric Tools: DGS/GATP/CAS/RGK/eLearning;
- Geometric Knowledge Management.

Synthetic Methods

Synthetic methods attempt to automate traditional geometry proof methods, producing human-readable proofs.

Seminal paper of Gelernter et al. It was based on the human simulation approach and has been considered a landmark in the AI area [Gel59, GHL60].

- Geometric reasoning - small and easy to understand proofs.
- Use of predicates only allow reaching fix-points.
- numerical model;
- constructing auxiliary points;
- generating geometric lemmas.

In spite of the success and significant improvements with these methods, the results did not lead to the development of a powerful geometry theorem prover [BdC95, CP79, CP86, Gil70, KA90, Nev74, Qua89]

Gelernter's GATP

A long-range program directed at the problem of "intelligent" behaviour and learning in machines has attained its first objective in the simulation on a high-speed digital computer of a machine capable of discovering proofs in elementary Euclidean plane geometry without resorting to exhaustive enumeration or to a decision procedure. The particular problem of a theorem proving in plane geometry was chosen as representative of a large class of difficult tasks that seem to require ingenuity and intelligence for their successful completion.

The theorem proving program relies upon heuristic methods to restrain if from generating proof sequences that do not have a high a priori probability of leading to a proof for the theorem in hand.

H. Gelernter1959, Realization of a geometry-theorem proving machine, Computers & thought, MIT Press, 1995

Gelernter's GATP

Backward chaining approach.

$$\forall_{\text{geometric elements}}[(H_1 \wedge \cdots \wedge H_r) \Rightarrow G]$$

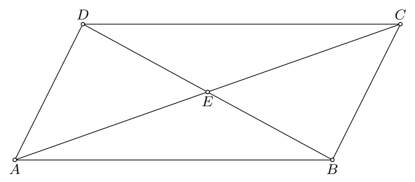
To prove G we search the *axiom rule set* to find a rule of the following form

$$[(G_1 \wedge \cdots \wedge G_r) \Rightarrow G]$$

until the sub-goals are hypothesis.

The proof search will generate an and-or-proof-tree.

Example 1 - Gelernter



$$points(A,B,C) \wedge AB \| CD \wedge AD \| BC \wedge coll(E,A,C) \wedge coll(E,B,D) \Rightarrow AE = EC$$

$\top$

$AC = CA$

$\top$ — $AB \| CD$

$\angle ACD = \angle CAB$

$\top$ — $AD \| BC$

$\angle CAD = \angle ACB$

$\Delta ABC \cong \Delta CDA$

$AB = CD$

$\top$

$\angle AEB = \angle CED$

$\top$ — $AB \| CD$

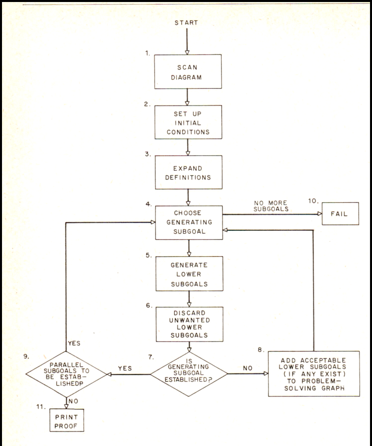
$\angle ACD = \angle CAB$ — $coll(E,A,C)$

$\angle ECD = \angle EAB$

$\Delta ECD \cong \Delta EAB$

$AE = EC$

Galernter (1959): Algorithm & Proof



PREMISES

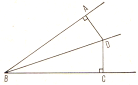
ANGLE ABD EQUALS ANGLE DBC
SEGMENT AD PERPEDICULAR SEGMENT AB
SEGMENT DC PERPEDICULAR SEGMENT BC

DEFINITION

RIGHT-ANGLE DAB
RIGHT-ANGLE DCB

SYNTACTIC SYMMETRIES

****
CA
BB
AC
DD
****



GOALS

SEGMENT AD EQUALS SEGMENT CD

Fig. 4.

**SOLUTION **

ANGLE ABD EQUALS ANGLE DBC
PREMISE
RIGHT-ANGLE DAB
DEFINTION OF PERPENDICULAR
RIGHT-ANGLE DCB
DEFINTION OF PERPENDICULAR
ANGLE BAD EQUALS ANGLE BCD
ALL RIGHT ANGLES ARE EQUAL
SEGMENT DB
ASSUMPTION BASED ON DIAGRAM
SEGMENT BD EQUALS SEGMENT BD
IDENTITY
TRIANGLE BCD
ASSUMPTION BASED ON DIAGRAM
TRIANGLE BAD
ASSUMPTION BASED ON DIAGRAM
TRIANGLE ADB CONGRUENT TRIANGLE CDB
SIDE-ANGLE-ANGLE
SEGMENT AD EQUALS SEGMENT CD
CORRESPONDING ELEMENTS OF CONGRUENT TRIANGLES ARE EQUAL

TOTAL ELAPSED TIME = 0.3200

GEOM — A "Coelho" out of the hat

Two uses of the geometric diagram as a model [CP86]:

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

Top-down or bottom-up directions? A general prover should be able to mix both directions of execution [CP86].

The introduction of new points can be envisaged as a means to make explicit more information in the model [CP86].

Although various strategies and heuristics were subsequently adopted and implement, the problem of search space explosion still remains and makes the methods of this type highly inefficient [CP79, CP86].

Example - GEOM

GEOM is a Prolog program that generates proofs for problems in high school plane geometry [CP86].

A user presents problems to GEOM by declaring the hypotheses, the optional diagram and the goal.

GEOM starts from the goal, top-down and with a depth-first strategy, outputing its deductions and reasons for each step of the proof.

The diagram works mostly as a source of counter-examples for pruning unprovable goals, and so proofs need not depend on it (...). However, the diagram may also be used in a positive guiding way.

Example - GEOM

The geometric knowledge of GEOM, i.e. some of the axioms and theorems of elementary plane geometry, is embodied in nine procedures.

They are: equal angles (EAI), right angles (RAI), equal magnitude (EM, EM1), equal segments (ESI), midpoints (MP), parallel segments (PRI), parallelogram (PG), congruence (DIRCON) and diagram routines.

Because each procedure may call itself through others, **the search space can grow quite large**, in particular when the clause for differences of segments is used.

GEOM: proof tree

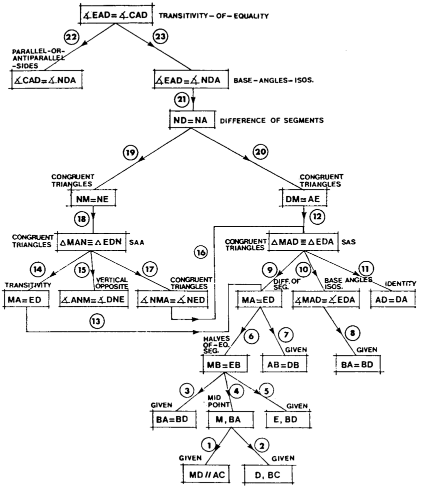


FIG. 22 — The proof tree of Problem 14 used by GEOM

Coordinate-free Methods

Instead of coordinates, some basic geometric quantities, e.g. the ratio of parallel line segments, the signed area, and the Pythagorean difference (vector methods).

- Area method [CGZ93, JNQ12, QJ06b];
- Full-angle method [CGZ94, CGZ96b];
- Solid geometry [CGZ95].

Pros: Geometric proofs, small and human-readable.

Cons:

- not the “normal” high-school geometric reasoning;
- for many conjectures these methods still deal with extremely complex expressions.

Area Method — Basic Geometric Quantities

Definition (Ratio of directed parallel segments)

For four collinear points P, Q, A, and B, such that $A \neq B$, the ratio of directed parallel segments, denoted $\frac{\overline{PQ}}{\overline{AB}}$ is a real number.

Definition (Signed Area)

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.

Definition (Pythagoras difference)

For three points A, B, and C, the *Pythagoras difference*, is defined in the following way: $\mathcal{P}_{ABC} = \overline{AB}^2 + \overline{CB}^2 - \overline{AC}^2$.

Properties of the Ratio of Directed Parallel Segments

- $\frac{\overline{PQ}}{\overline{AB}} = -\frac{\overline{QP}}{\overline{AB}} = \frac{\overline{QP}}{\overline{BA}} = -\frac{\overline{PQ}}{\overline{BA}}$;
- $\frac{\overline{PQ}}{\overline{AB}} = 0$ iff $P = Q$;
- $(\ldots)$

EL1 (The Co-side Theorem) Let M be the intersection of two non-parallel lines AB and PQ and $Q \neq M$. Then it holds that $\frac{\overline{PM}}{\overline{QM}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{QAB}}$; $\frac{\overline{PM}}{\overline{PQ}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAQB}}$; $\frac{\overline{QM}}{\overline{PQ}} = \frac{\mathcal{S}_{QAB}}{\mathcal{S}_{PAQB}}$.

Properties of the Signed Area

- $\mathcal{S}_{ABC} = \mathcal{S}_{CAB} = \mathcal{S}_{BCA} = -\mathcal{S}_{ACB} = -\mathcal{S}_{BAC} = -\mathcal{S}_{CBA}$.
- $\mathcal{S}_{ABC} = 0$ iff A, B, and C are collinear.
- $PQ \parallel AB$ iff $\mathcal{S}_{PAB} = \mathcal{S}_{QAB}$, i.e., iff $\mathcal{S}_{PAQB} = 0$.
- Let $ABCD$ be a parallelogram, P and Q be two arbitrary points. Then it holds that $\mathcal{S}_{APQ} + \mathcal{S}_{CPQ} = \mathcal{S}_{BPQ} + \mathcal{S}_{DPQ}$ or $\mathcal{S}_{PAQB} = \mathcal{S}_{PDQC}$.
- Let R be a point on the line PQ. Then for any two points A and B it holds that $\mathcal{S}_{RAB} = \frac{\overline{PR}}{\overline{PQ}}\mathcal{S}_{QAB} + \frac{\overline{RQ}}{\overline{PQ}}\mathcal{S}_{PAB}$.
- $(\ldots)$

Properties of the Pythagoras Difference

- $\mathcal{P}_{AAB} = 0$.
- $\mathcal{P}_{ABC} = \mathcal{P}_{CBA}$.
- If A, B, and C are collinear then, $\mathcal{P}_{ABC} = 2\overline{BA}\,\overline{BC}$.
- $AB \perp BC$ iff $\mathcal{P}_{ABC} = 0$.
- Let AB and PQ be two non-perpendicular lines, and Y be the intersection of line PQ and the line passing through A and perpendicular to AB. Then, it holds that

$$\frac{\overline{PY}}{\overline{QY}} = \frac{\mathcal{P}_{PAB}}{\mathcal{P}_{QAB}}, \quad \frac{\overline{PY}}{\overline{PQ}} = \frac{\mathcal{P}_{PAB}}{\mathcal{P}_{PAQB}}, \quad \frac{\overline{QY}}{\overline{PQ}} = \frac{\mathcal{P}_{QAB}}{\mathcal{P}_{PAQB}}.$$

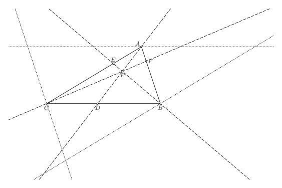
- (...)

CISUC

The Area Method — The Proof Algorithm

Express the hypothesis of a theorem using a set of constructive statements.

Each constructive statement introduces a new point.



The conclusion is expressed by a polynomial in some geometry quantities (defined above), without any relation to a given system of coordinates.

The proof is then developed by eliminating, in reverse order, the point introduced before, using for that purpose a set of lemmas.

Constructive Geometric Statements

ECS1 construction of an arbitrary point U; (. . .).

ECS2 construction of a point Y such that it is the intersection of two lines (LINE U V) and (LINE P Q);
ndg-condition: $UV \nparallel PQ$; $U \neq V$; $P \neq Q$.
degree of freedom for Y: 0

ECS3 construction of a point Y such that it is a foot from a given point P to (LINE U V); (. . .).

ECS4 construction of a point Y on the line passing through point W and parallel to (LINE U V), such that $\overline{WY} = r\overline{UV}$, (. . .).

ECS5 construction of a point Y on the line passing through point U and perpendicular to (LINE U V), such that $r = \frac{4\mathcal{S}_{UVY}}{\mathcal{P}_{UVU}}$, (. . .).

Forms of Expressing the Conclusion

property	in terms of geometric quantities
points A and B are identical	$\mathcal{P}_{ABA} = 0$
points A, B, C are collinear	$\mathcal{S}_{ABC} = 0$
AB is perpendicular to CD	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{CDC} \neq 0 \wedge \mathcal{P}_{ACD} = \mathcal{P}_{BCD}$
AB is parallel to CD	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{CDC} \neq 0 \wedge \mathcal{S}_{ACD} = \mathcal{S}_{BCD}$
O is the midpoint of AB	$\mathcal{S}_{ABO} = 0 \wedge \mathcal{P}_{ABA} \neq 0 \wedge \frac{\overline{AO}}{\overline{AB}} = \frac{1}{2}$
AB has the same length as CD	$\mathcal{P}_{ABA} = \mathcal{P}_{CDC}$
points A, B, C, D are harmonic	$\mathcal{S}_{ABC} = 0 \wedge \mathcal{S}_{ABD} = 0 \wedge \mathcal{P}_{BCB} \neq 0 \wedge \mathcal{P}_{BDB} \neq 0 \wedge \frac{\overline{AC}}{\overline{CB}} = \frac{\overline{DA}}{\overline{DB}}$
angle ABC has the same measure as DEF	$\mathcal{P}_{ABA} \neq 0 \wedge \mathcal{P}_{ACA} \neq 0 \wedge \mathcal{P}_{BCB} \neq 0 \wedge \mathcal{P}_{DED} \neq 0 \wedge \mathcal{P}_{DFD} \neq 0 \wedge \mathcal{P}_{EFE} \neq 0 \wedge \mathcal{S}_{ABC} \cdot \mathcal{P}_{DEF} = \mathcal{S}_{DEF} \cdot \mathcal{P}_{ABC}$
A and B belong to the same circle arc CD	$\mathcal{S}_{ACD} \neq 0 \wedge \mathcal{S}_{BCD} \neq 0 \wedge \mathcal{S}_{CAD} \cdot \mathcal{P}_{CBD} = \mathcal{S}_{CBD} \cdot \mathcal{P}_{CAD}$

Elimination Lemmas

EL2 Let $G(Y)$ be a linear geometric quantity and point Y is introduced by the construction (PRATIO Y W (LINE U V) r). Then it holds

$$G(Y) = G(W) + r(G(V) - G(U)).$$

EL3 Let $G(Y)$ be a linear geometric quantity and point Y is introduced by the construction (INTER Y (LINE U V) (LINE P Q). Then it holds

$$G(Y) = \frac{\mathcal{S}_{UPQ}G(V) - \mathcal{S}_{VPQ}G(U)}{\mathcal{S}_{UPVQ}}.$$

- (...)

Constructive Steps & Elimination Lemmas

		Geometric Quantities						
		$\mathcal{P}_{AYB}$	$\mathcal{P}_{ABY}$	$\mathcal{P}_{ABCY}$	$\mathcal{S}_{ABY}$	$\mathcal{S}_{ABCY}$	$\frac{\overline{AY}}{\overline{CD}}$	$\frac{\overline{AY}}{\overline{BY}}$
Constructive Steps	ECS2	EL5	EL3				EL11	EL1
	ECS3	EL6	EL4				EL12	
	ECS4	EL7	EL2				EL13	
	ECS5	EL10	EL9		EL8		EL14	
		Elimination Lemmas						

The Algorithm

→ $S = (C_1, C_2, \ldots, C_m, (E, F))$ is a statement in **C**.

← The algorithm tells whether S is true, or not, and if it is true, produces a proof for S.

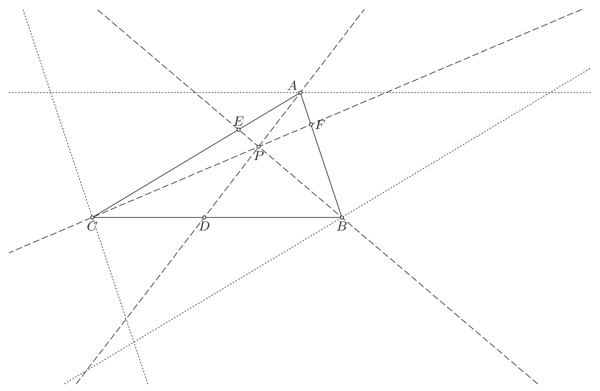
```
for (i=m;i==1;i--) {
   if (the ndg conditions of Ci is satisfied) exit;
   // Let G1,\ldots,Gn be the geometric quantities in E and F
   for (j=1;j<=n,j++) {
      Hj <- eliminating the point introduced
                        by construction Ci from Gj
      E <- E[Gj:=Hj]
      F <- F[Gj:=Hj]
   }
}
if (E==F) S <- true else S<-false
```

Adding to that it is needed to check the ndg condition of a construction (three possible forms).

An Example (Ceva's Theorem)

Let $\triangle ABC$ be a triangle and P be an arbitrary point in the plane. Let D be the intersection of AP and BC, E be the intersection of BP and AC, and F the intersection of CP and AB. Then:

$$\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} = 1$$



Example — Proof

The proof of a conjecture is based on eliminating all the constructed points, in reverse order, until an equality in only the free points is reached.

$$
\begin{aligned}
\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} &= \frac{\mathcal{S}_{APC}}{\mathcal{S}_{BCP}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} && \text{the point } F \text{ is eliminated}\\
&= \frac{\mathcal{S}_{APC}}{\mathcal{S}_{BCP}}\frac{\mathcal{S}_{BPA}}{\mathcal{S}_{CAP}}\frac{\overline{CE}}{\overline{EA}} && \text{the point } D \text{ is eliminated}\\
&= \frac{\mathcal{S}_{APC}}{\mathcal{S}_{BCP}}\frac{\mathcal{S}_{BPA}}{\mathcal{S}_{CAP}}\frac{\mathcal{S}_{CPB}}{\mathcal{S}_{ABP}} && \text{the point } E \text{ is eliminated}\\
&= 1
\end{aligned}
$$

Elimination Steps: 3; Geometric Steps: 6; Algebraic Steps: 23; Total Steps: 32; CPU Time 0.004s. The GATP also provide the ndg conditions.

Full-Angle Method

Intuitively, a full-angle $\angle[u, v]$ is the angle from line u to line v. Two full-angles $\angle[l, m]$ and $\angle[u, v]$ are equal if there exists a rotation K such that $K(l) \| u$ and $K(m) \| v$

Full-Angle is defined as an ordered pair of lines which satisfies the following rules [CGZ96b]:

- R1 For all parallel lines $AB \| PQ$, $\angle[0] = \angle[AB, PQ]$ is a constant.
- R2 For all perpendicular lines $AB \perp PQ$, $\angle[1] = \angle[AB, PQ]$ is a constant.
- R7 If PX is parallel to UV, then $\angle[AB, PX] = \angle[AB, UV]$.
- R8 If PX is perpendicular to UV, then $\angle[AB, PX] = \angle[1] + \angle[AB, UV]$.

Solid Geometry

Solid Geometry Method — For any points A, B, C and D in the space, the signed volume V_{ABCD} of the tetrahedron $ABCD$ is a real number which satisfies the following properties [CGZ95].

V.1 When two neighbor vertices of the tetrahedron are interchanged, the signed volume of the tetrahedron will change signs, e.g., $V_{ABCD} = -V_{ABDC}$.

V.2 Points A, B, C and D are coplanar iff $V_{ABCD} = 0$.

V.3 There exist at least four points A, B, C and D such that $V_{ABCD} \neq 0$.

V.4 For five points A, B, C, D and O, we have $V_{ABCD} = V_{ABCO} + V_{ABOD} + V_{AOCD} + V_{OBCD}$.

V.5 If A, B, C, D, E and F are six coplanar points and $S_{ABC} = \lambda SDEF$ then for any point T we have $V_{TABC} = \lambda V_{TDEF}$.

Algebraic Methods

Algebraic Methods: are based on reducing geometry properties to algebraic properties expressed in terms of Cartesian coordinates.

The biggest successes in automated theorem proving in geometry were achieved (i.e., the most complex theorems were proved) by algebraic provers based on:

- Wu's method [Cho85, Cho88];
- Gröbner bases method [Buc98, Kap86].

Decision procedures.

No readable, traditional geometry proofs, only a yes/no answer (with a corresponding algebraic argument).

Wu's Method

An elementary version of Wu's method is simple:
Geometric theorem T transcribed as polynomial equations and inequations of the form:

- H: $h_l = O, \ldots, h_s = O, d_1 \neq 0, \ldots, d_t \neq 0$;
- C: c=0.

Proving T is equivalent to deciding whether the formula

$$\forall_{x_l,\ldots x_n}[h_1 = 0 \wedge \cdots \wedge h_s \wedge d_1 \neq 0 \wedge \ldots \wedge d_t \neq 0 \Rightarrow c = 0] \quad (1)$$

is valid.

Wu's Method

Computes a characteristic set C of $\{h_1, \ldots, h_s\}$ and the pseudo-remainder r of c with respect to C.

If r is identically equal to 0, then T is proved to be true.

The subsidiary condition $J \neq 0$, where J is the product of initials of the polynomials in C are the ndg conditions [CG90, WT86, Wu00].

This is a decision procedure.

GCLC Implementation of Wu's Method

Let $\triangle ABC$ be a triangle and P be an arbitrary point in the plane. Let D be the intersection of AP and BC, E be the intersection of BP and AC, and F the intersection of CP and AB. Then: $\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} = 1$

$$
\begin{aligned}
p_1 &= -u_3x_2 + (u_2 - u_1)x_1 + u_3u_1\\
p_2 &= u_5x_2 - u_4x_1\\
p_3 &= -u_3x_4 + u_2x_3\\
p_4 &= u_5x_4 + (-u_4 + u_1)x_3 - u_5u_1\\
p_5 &= (u_5 - u_3)x_6 + (-u_5u_2 + u_4u_3)\\
p_6 &= 2x_6x_3^2x_1^3 - 3u_3x_6x_3^2x_1^2 + u_3^2x_6x_3^2x_1 - u_3x_6x_3x_1^3 + u_3^2x_6x_3x_1^2 -\\
&\quad -u_1x_3^2x_1^3 + 2u_3u_1x_3^2x_1^2 - u_3^2u_1x_3^2x_1
\end{aligned}
$$

CISUC

GCLC Implementation of Wu's Method (cont)

Triangulation, step 1; step 2; step 3; step 4; step 5

Calculating final remainder of the conclusion:
$g = 2x_6x_3^2x_1^3 - 3u_3x_6x_3^2x_1^2 + u_3^2x_6x_3^2x_1 - u_3x_6x_3x_1^3 + u_3^2x_6x_3x_1^2 - u_1x_3^2x_1^3 + 2u_3u_1x_3^2x_1^2 - u_3^2u_1x_3^2x_1$
with respect to the triangular system.

Pseudo remainder with p_4 over variable x_6:
$g = (2u_5u_2 - u_5u_1 - 2u_4u_3 + u_3u_1)x_3^2x_1^3 + (-3u_5u_3u_2 + 2u_5u_3u_1 + 3u_4u_3^2 - 2u_3^2u_1)x_3^2x_1^2 + (u_5u_3^2u_2 - u_5u_3^2u_1 - u_4u_3^3 + u_3^3u_1)x_3^2x_1 + (-u_5u_3u_2 + u_4u_3^2)x_3x_1^3 + (u_5u_3^2u_2 - u_4u_3^3)x_3x_1^2$

$(\ldots)$

Pseudo remainder with p_0 over variable x_1: $g = 0$

Status: The conjecture has been proved.

...but all the calculations made, are not translatable to geometric reasoning.

Gröbner Basis

A Gröbner basis of an ideal is a special basis using which the membership problem of the ideal as well as the membership problem of the radical of the ideal can be easily decided.

(. . .) to decide whether a finite set of geometry hypotheses expressed as polynomial equations, in conjunction with a finite set of subsidiary conditions expressed as negations of polynomial equations which rule out degenerate cases, imply another geometry relation given as a conclusion.

Such a problem is shown to be equivalent to deciding whether a finite set of polynomials does not have a solution in an algebraically closed field. Using Hilbert's Nullstellensatz, this problem can be decided by checking whether 1 is in the ideal generated by these polynomials

This test can be done by computing a Gröbner basis of the ideal.

GCLC Implementation of Gröbner Basis Method

Let $\triangle ABC$ be a triangle and P be an arbitrary point in the plane. Let D be the intersection of AP and BC, E be the intersection of BP and AC, and F the intersection of CP and AB. Then: $\frac{\overline{AF}}{\overline{FB}} \frac{\overline{BD}}{\overline{DC}} \frac{\overline{CE}}{\overline{EA}} = 1$.

Conjecture $p_6 = 2x_6x_3^2x_1^3 - 3u_3x_6x_3^2x_1^2 + u_3^2x_6x_3^2x_1 - u_3x_6x_3x_1^3 + u_3^2x_6x_3x_1^2 - u_1x_3^2x_1^3 + 2u_3u_1x_3^2x_1^2 - u_3^2u_1x_3^2x_1$

The used proving method is Buchberger's method.
Input polynomial system is:

$$
\begin{aligned}
p_0 &= -u_3x_2 + (u_2 - u_1)x_1 + u_3u_1\\
p_1 &= u_5x_2 - u_4x_1\\
p_2 &= -u_3x_4 + u_2x_3\\
p_3 &= u_5x_4 + (-u_4 + u_1)x_3 - u_5u_1\\
p_4 &= (u_5 - u_3)x_6 + (-u_5u_2 + u_4u_3)
\end{aligned}
$$

GCLC Implementation of Gröbner Basis Method (cont)

iteration 1; iteration 2.

Gröbner basis has 7 polynomials:

$p_0 = -u_3x_2 + (u_2 - u_1)x_1 + u_3u_1$

$p_1 = u_5x_2 - u_4x_1$

$p_2 = -u_3x_4 + u_2x_3$

$p_3 = u_5x_4 + (-u_4 + u_1)x_3 - u_5u_1$

$p_4 = (u_5 - u_3)x_6 + (-u_5u_2 + u_4u_3)$

$p_5 = (u_5u_2 - u_5u_1 - u_4u_3)x_1 + u_5u_3u_1$

$p_6 = (u_5u_2 - u_4u_3 + u_3u_1)x_3 - u_5u_3u_1$

$(\ldots)$

Status: The conjecture has been proved.
Space Complexity: The biggest polynomial obtained during proof process contained 259 terms.
Time Complexity: Time spent by the prover is 0.101 seconds.

"New" approaches

- An approach based on a deductive database and forward chaining works over a suitably selected set of higher-order lemmas and can prove complex geometry theorems, but still has a smaller scope than algebraic provers [CGZ94, CGZ00, YCG10b].
- Quaife used a resolution theorem prover to prove theorems in Tarski's geometry [Qua89].
- A GATP based on coherent-logic capable of producing both readable and formal proofs of geometric conjectures of certain sort [SPJ11].
- Probabilistic verification of elementary geometry statements [CFGG97, RGK99].
- Visual Reasoning/Proofs [Kim89, YCG10a, YCG10b].

Geometry Deductive Database

- In the general setting: structured deductive database and the data-based search strategy to improve the search efficiency.[1]
- Selection of a good set of rules; adding auxiliary points and constructing numerical diagrams as models automatically.

The result program can be used to find fix-points for a geometric configuration, i.e. the program can find all the properties of the configuration that can be deduced using a fixed set of geometric rules.

Generate ndg conditions.

Structured deductive database (graphs) reduce the size of the database in some cases by one thousand times.

[1] Semantic Graphs are an alternative!?

Deductive Databases

- Use canonical form for predicates;
- Use equivalent classes to represent some predicates;
- Use representative elements for equivalent classes;
- breadth-first forward chaining search:
  where D_0 is the hypotheses of the geometry statement and R is the rule set.

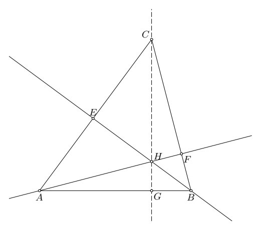
  For each rule r in R, apply it to D_0 to obtain new facts. Let D_1 be the union of D_0 and the set of new facts obtained.

  Repeat the above process for D_1 to obtain D_2, and so on.

  If at certain step $D_k = D_{k+1}$, we say that a fix-point for D_0 and R is reached.

Geometry Deductive Database – The Orthocenter Theorem

$points(A, B, C) \wedge coll(E, A, C) \wedge perp(B, E, A, C) \wedge coll(F, B, C) \wedge perp(A, F, B, C) \wedge coll(H, A, F) \wedge coll(H, B, E) \wedge coll(G, A, B) \wedge coll(G, C, H)$



The fix-point contains two of the most often encountered properties of this configuration:

- $perp(C, G, A, B)$;
- $\angle FGC = \angle CGE$

Quaife's GATP

Tarski axiomatic system: is, or rather its algebraic equivalent, complete and decidable.

Quaife developed a GATP for Euclidean plane geometry within the automated reasoning system OTTER (a resolution theorem prover) [Qua89].

(A1) Reflexivity axiom for equidistance.
$\rightarrow u \cdot v \equiv v \cdot u$

(A2) Transitivity axiom for equidistance.
$u \cdot v \equiv w \cdot x, u \cdot v \equiv y \cdot z \rightarrow w \cdot x \equiv y \cdot z$

(A4) Segment construction axiom, two clauses.
(A4.1) $\rightarrow B(u, v, Ext(u, v, w, x))$
(A4.2) $\rightarrow v \cdot Ext(u, v, w, x) \equiv w \cdot x$

(...)

Quaife's GATP

Heuristics

- maximum weight for retained clauses at 25,
- first attempt to obtain a proof in which no variables are allowed in any generated and retained clause.

The provers based upon Wu's algorithm, are able to prove quite more difficult theorems in geometry then those by Quaife's GATP.

However Wu's method only works with hypotheses and theorems that can be expressed as equations, and not with inequalities as correspond to the relation B in Quaife's resolution prover.

Coherent Logic GATP

Coherent Logic is a fragment of first-order logic with formulae of the following form:

$$A_1(x) \wedge \ldots \wedge A_n(x) \rightarrow \exists_{y_1} B(x, y_1) \vee \ldots \vee \exists_{y_m} B(x, y_m)$$

with a breath-first proof procedure sound and complete [BC05].

ArgoCLP (**C**oherent **L**ogic **P**rover of the Argo Group[2])

- new proof procedures;
- proof trace exportable to:
  - a proof object in Isabelle/Isar;
  - human readable (English/LaTeX).

not aimed at proving complex geometry theorems but rather at proving foundational theorems (close to the axiom level) [SPJ11].

[2] http://argo.matf.bg.ac.rs/

Probabilistic Verification

Probabilistic verification of elementary geometry statements [CFGG97, RGK99].

> *Cinderella (...) use (...) a technique called "Randomized Theorem Checking". First the conjecture (...) is generated. Then the configuration is moved into many different [random] positions and for each of these it is checked whether the conjecture still holds. (...) generating enough(!) random (!) examples where the theorem holds is at least as convincing as a computer-generated symbolic proof.*
>
> *User Manual for the Interactive Geometry Software Cinderella, Jürgen Richter-Gebert, Ulrich H. Kortenkamp*

Visual Reasoning/Representation

Visual Reasoning extend the use of diagrams with a method that allows the diagrams to be perceived and to be manipulated in a creative manner [Kim89].

Visually Dynamic Presentation of Proofs linking the proof done by a synthetic method (full-angle) with a visual presentation of the proof [QSGB19, SQ10, YCG10a, YCG10b].

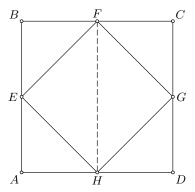
Visual Reasoning in Geometry Theorem Proving

We study the role of visual reasoning as a computationally feasible heuristic tool in geometry problem solving. We use an algebraic notation to represent geometric objects and to manipulate them.
We show that this representation captures powerful heuristics for proving geometry theorems, and that it allows a systematic manipulation of geometric features in a manner similar to what may occur in human visual reasoning

Michelle Y . Kim,

An Example

Consider the problem in "Given a square $ABCD$, take the midpoints of the four sides, and prove that the two triangles ΔEEH and ΔGFH are congruent to each other."



To solve this problem, backward-chaining methods used by most of previous geometry-theorem proving systems [Gel59, CP86] would first set up a goal to prove that the two triangles arc congruent (...). A human mathematician, given the problem , may perceive an apparent symmetry in the diagram by observing a reflection across FH or across EG. As a symmetry is observed, it can be shown with little effort that the two triangles are congruent, and thus repeated proofs can be avoided.

Visually Dynamic Presentation of Proofs in Plane Geometry

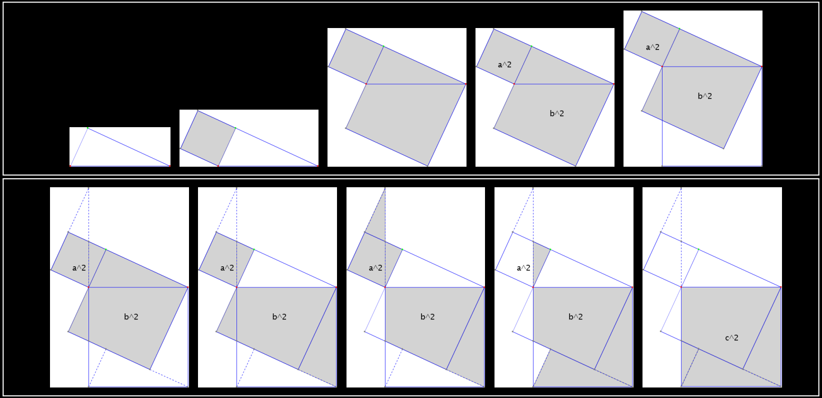


Figure: Pythagoras Theorem - Visual Proof

Formalisation

Full formal proofs mechanically verified by generic theorem proof assistants (e.g. Isabelle [Pau94, PN90], Coq [Tea09]).

- Hilbert's *Foundations of Geometry* [Hil77, MF03, DDS00];
- Jan von Plato's constructive geometry [Kah95, vP95];
- French high school geometry [Gui04];
- Tarski's geometry [Nar07b, BBN16];
- An axiom system for compass and ruler geometry [BNW18];
- Projective geometry [MNS11, FT11];
- Area Method [JNQ12, Nar06];
- Algebraic methods in geometry [MPPJ12].

Proof Assistants

Proof assistant (or interactive theorem prover) is a software tool to assist with the development of formal proofs by human-machine collaboration.

- Isabelle—`https://isabelle.in.tum.de/`—Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus.
- Coq—`https://coq.inria.fr/`—Coq is a formal proof management system. It provides a formal language to write mathematical definitions, executable algorithms and theorems together with an environment for semi-interactive development of machine-checked proofs.

Others: HOL Light; Lean; Mizar; ...

Area Method: Formalisation

Formalisation [JNQ12, Nar06, Nar09];

1. $\overline{AB} = 0$ if and only if the points A and B are identical
2. $\mathcal{S}_{ABC} = \mathcal{S}_{CAB}$
3. $\mathcal{S}_{ABC} = -\mathcal{S}_{BAC}$
4. If $\mathcal{S}_{ABC} = 0$ then $\overline{AB} + \overline{BC} = \overline{AC}$ (Chasles's axiom)
5. There are points A, B, C such that $\mathcal{S}_{ABC} \neq 0$ (dimension; not all points are collinear)
6. $\mathcal{S}_{ABC} = \mathcal{S}_{DBC} + \mathcal{S}_{ADC} + \mathcal{S}_{ABD}$ (dimension; all points are in the same plane)
7. For each element r of F, there exists a point P, such that $\mathcal{S}_{ABP} = 0$ and $\overline{AP} = r\overline{AB}$ (construction of a point on the line)
8. If $A \neq B, \mathcal{S}_{ABP} = 0, \overline{AP} = r\overline{AB}, \mathcal{S}_{ABP'} = 0$ and $\overline{AP'} = r\overline{AB}$, then $P = P'$ (unicity)
9. If $PQ \parallel CD$ and $\frac{\overline{PQ}}{\overline{CD}} = 1$ then $DQ \parallel PC$ (parallelogram)
10. If $\mathcal{S}_{PAC} \neq 0$ and $\mathcal{S}_{ABC} = 0$ then $\frac{\overline{AB}}{\overline{AC}} = \frac{\mathcal{S}_{PAB}}{\mathcal{S}_{PAC}}$ (proportions)
11. If $C \neq D$ and $AB \perp CD$ and $EF \perp CD$ then $AB \parallel EF$
12. If $A \neq B$ and $AB \perp CD$ and $AB \parallel EF$ then $EF \perp CD$
13. If $FA \perp BC$ and $\mathcal{S}_{FBC} = 0$ then $4\mathcal{S}^2_{ABC} = \overline{AF}^2\overline{BC}^2$ (area of a triangle)

Using this axiom system all the properties of the geometric quantities required by the area method were *formally verified* (within the *Coq* proof assistant), demonstrating the correctness of the system and eliminating all concerns about provability of the lemmas [Nar09].

Area Method: Formalization in Coq

```
Require Export field.
Require Import Classical.

Ltac Geometry := auto with Geom field_hints.

Parameter Point : Set. // The set of Points
Parameter S : Point -> Point -> Point -> F. // The signed area
Parameter DSeg : Point -> Point -> F. // The signed distance

Infix "**" := DSeg (left associativity, at level 20) : F_scope.

Definition Col (A B C : Point) : Prop := S A B C = 0.
Definition S4 (A B C D : Point) : F := S A B C + S A C D.
Definition parallel (A B C D : Point) : Prop := S4 A C B D = 0.

Axiom A1b : forall A B : Point, A ** B = 0 <-> A = B.

Axiom A2a : forall (A B : Point) (r : F),
    {P : Point | Col A B P /\ A ** P = r * A ** B}.
Axiom A2b : forall (A B P P1 : Point) (r : F),
    A <> B ->
    Col A B P ->
    A ** P = r * A ** B -> Col A B P1 -> A ** P1 = r * A ** B -> P = P1.

Axiom chasles : forall A B C : Point, Col A B C -> A ** B + B ** C = A ** C.
```

CISUC

Automated Discovery

- Locus Generation: to determine the implicit equation of a locus set [BAE07, BA12].

  The set of points determined by the different positions of a point, the *tracer*, as a second point in which the tracer depends on, called the *mover*, runs along the one dimensional object to which it is restrained.

- Automated Finding of Theorems: the discovery of new facts about a given geometric configuration.

Finding locus equations

For most DGS a locus is basically a set of points in the screen with no algebraic information [BAE07, ABMR14].

- Numerical approach, based on interpolation (Cinderella, Cabri) [Bot02].
- Symbolic method, finding the equation of a locus [BL02, BA12, ABMR14].

Determine the equation of a locus set using remote computations on a server [EBA10].

Loci Finding: Algorithm

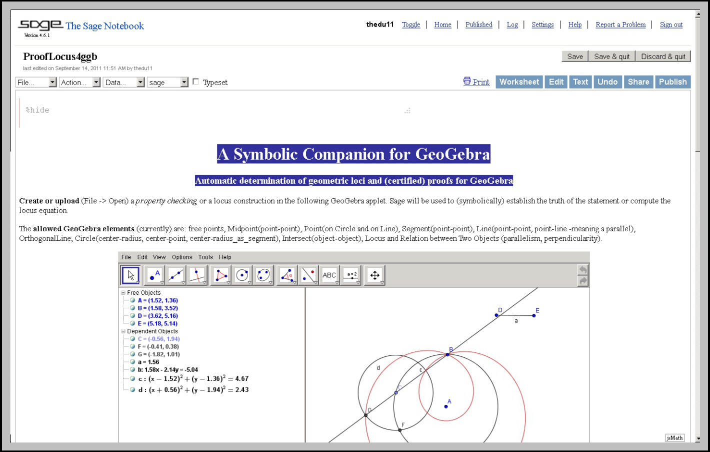
A statement is considered where the conclusion does not follow from the hypotheses.

Symbolic coordinates are assigned to the points of the construction (where every free point gets two new free variables u_i , u_{i+1} , and every bounded point gets up to two new dependent variables x_j, x_{j+1}) so the hypotheses and thesis are rewritten as polynomials $h_1, \ldots, h_n$ and $t$in $\mathbb{Q}[u, x]$.

Eliminating the dependent variables in the ideal $(hypotheses, thesis)$, the vanishing of every element in the elimination ideal $(hypotheses, thesis) \cap \mathbb{Q}[u]$ is a necessary condition for the statement to hold.

Locus Finding: Implementation

A *Sage* worksheet integrating *GeoGebra*



Implementation (cont.)

Two different tasks are performed over GeoGebra constructions:

- the computation of the equation of a geometric locus in the case of a locus construction;

  `LocusEquation( <Locus Point>, <Moving Point> )`

- the study of the truth of a general geometric statement included in the GeoGebra construction as a Boolean variable.

Both tasks are implemented using algebraic automatic deduction techniques based on Gröbner bases computations.

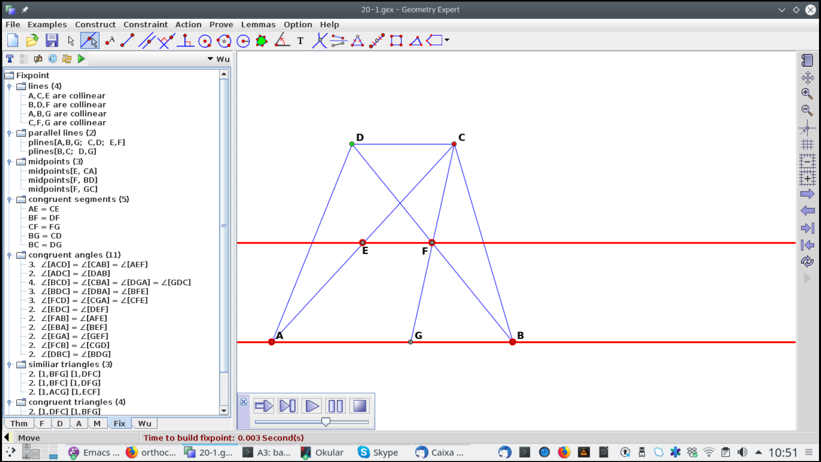
The algorithm, based on a recent work on the Gröbner cover of parametric systems, identifies degenerate components and extraneous adherence points in loci, both natural byproducts of general polynomial algebraic methods [BA12].

Automated Finding of Theorems

Deductive Database Approach. Forward chaining till reaching a fixed point.

> *An interesting application is to discover 'new' facts about a given geometric configuration.*
> *Our experiments show that our program can discover most of the well-known results and often some unexpected ones.*
> *Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. A deductive database approach to automated geometry theorem proving and discovering.*

JGEX: Automated Finding of Theorems



JGEX - example Manual, 20-1

Automated Geometer

The Automated Geometer, AG, (also meaning Amateur Geometer) intends to be a GeoGebra module where pure automatic discovery is performed.

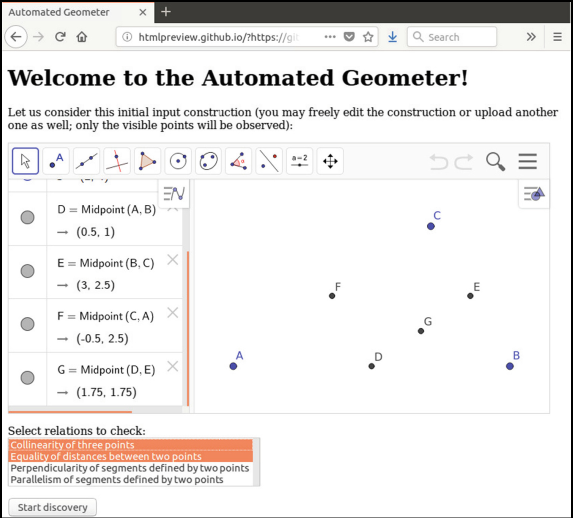
It includes a generator of further geometric elements from those of a given construction, and a set of rules for producing conjectures on the whole set of elements.

But the ultimate AG aim is not just performing a systematic exploration of the space of possible conjectures, but mimicking human thought when doing elementary geometry.

Francisco Botana, Zoltan Kovacs, and Tomas Recio. Towards an automated geometer. AISC 2018, LNCS 11110, Springer, 2018.

CISUC

Automated Geometer / Amateur Geometer



http://prover-test.geogebra.org/~kovzol/ag/automated-geometer.html

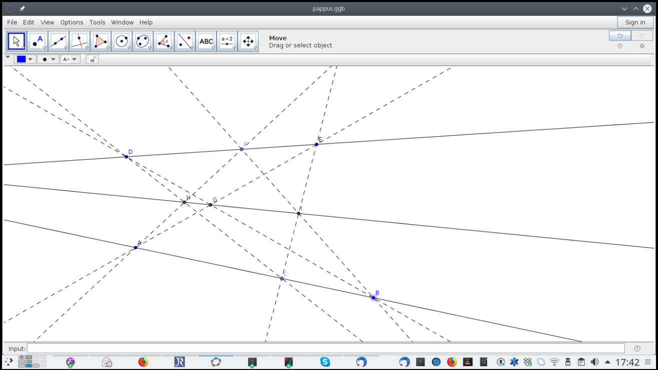
Geometric Tools

Geometric tools: Dynamic Geometry Software (DGS) & Geometry Automated Theorem Provers (GATP) & Computer Algebra Systems (CAS) & Repositories of Geometric Knowledge (RGK) & eLearning in Geometry.

- DGS: Cabri Geometry; C.a.R.; Cinderella; GCLC; GeoGebra; The Geometer's Sketchpad;
  JGEX [Gro11, CGY04, Hoh02, Jac01, Jan06, LS90, RGK99]; . . .
- GATP; GCLC; OpenGeoProver; JGEX; GeoProof; . . .
  - verification of the soundness of a geometric construction [JQ07].
  - reason about a given DGS construction [CGZ96a, JQ06, Nar07a, QJ06b].
  - human-readable proofs [JNQ12, QJ06a].
- RGK [QJ07, Qua11].
- eLearning [ABY86, HLY86, QJ06b, SQ08, QSM18, SQMC18]

Dynamic Geometry Software

DGS are computer environments which allow one to create and then manipulate geometric constructions, primarily in plane geometry.



Geometry Automated Theorem Provers: GCLC

Proving geometrical theorems by computer programs.

```
*** Ceva s theorem
point A 80 10
point B 50 80
point C 100 80
point P 75 65
line a B C
line b A C
line c A B
line pa P A
line pb P B
line pc P C
*** constructed point
intersec D a pa
intersec E b pb
intersec F c pc
*** conjecture
prove {equal{mult{mult{sratio A F F B}{sratio B D D C}}{sratio C E E A}}1}
```

CISUC

Geometry Automated Theorem Provers: GCLC

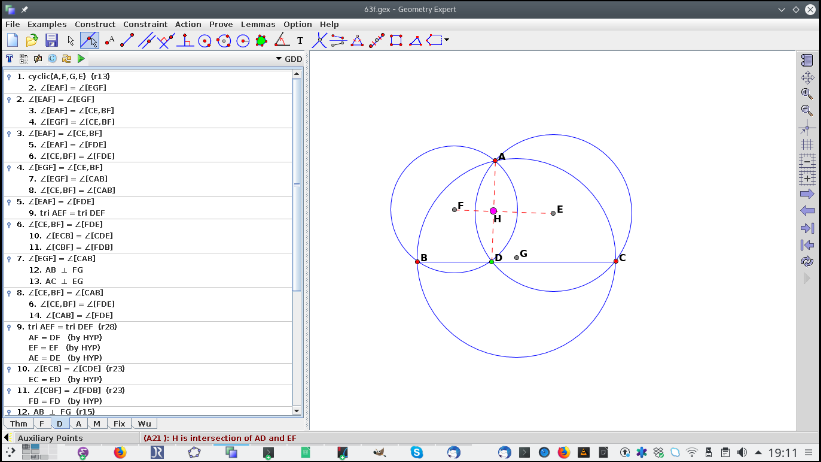
A5exemplos: bash — Konsole

Ficheiro Editar Ver Favoritos Configuração Ajuda

```
pedro@nomada:A5exemplos$ gclc ceva.gcl

GCLC 2015 (GC language (R) -> LaTeX  Converter)
Copyright (c) 1996-2015 by Predrag Janicic, University of Belgrade.
Licence Creative Commons CC BY-ND.

Input file: ceva.gcl
Output file: ceva.pic

Starting point number: 1

The theorem prover based on the area method used.
Number of elimination proof steps:      3
Number of geometric proof steps:        6
Number of algebraic proof steps:       23
Total number of proof steps:           32

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

File 'ceva.gcl' successfully processed.
Ending point number: 151

Transcript written on gclc.log.

pedro@nomada:A5exemplos$
```

CISUC

Geometry Automated Theorem Provers: JGEX



JGEX - GDD-FULL 63f

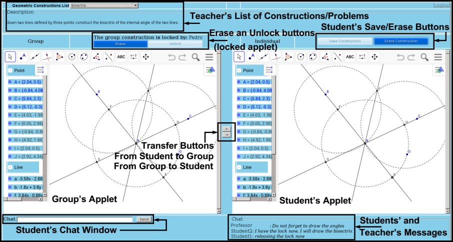
Integration: DGSs & GATPs

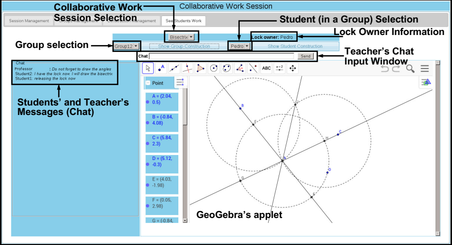
- GCLC/WinGCLC - A DGS tool that integrates three GATPs: Area Method, Wu's Method and Gröbner Bases Method [JQ06, Jan06].
- JGEX - is a software which combines a DGS and some GATPs (full angle, Wu's Method, Deductive Databases for the full angle) [YCG10a, YCG10b, CGY04].
- GeoProof - DGS tool that integrates three GATPs Area Method, Wu's Method and Gröbner Bases Method [Nar07a].
- GeoGebra - DGS + CAS + GATPs [ABK+16, BHJ+15, Kov15].
- Theorema Project - Theorema is a project that aims at supporting the entire process of mathematical theory exploration within one coherent logic and software system [BCJ+06]. Implementation of the Area Method[Rob02, Rob07].

Others: The Geometry Tutor, Mentoniezh, Defi, Chypre, Cabri-Euclide, Geometrix, Baghera, MMP-Geometer, Geometry Explorer, Cinderella.

Integration/eLearning (DGSs & GATPs & RGPs)

WebGeometryLab: A Web environment incorporating a DGS (GATPs) and a repository of geometric problems, that can be used in a synchronous and asynchronous fashion and with some adaptive and collaborative features. [QSM18, SQ08, SQMC18].





Others: Tabulae [MSB05]; GeoThink [MSM08]; Advanced Geometry Tutor [MV05]; AgentGeom [CFPR07]; geogebraTUTOR [RFHG07].

Integration Issues

Integrate a mosaic of tools into a coherent system.

- Intergeo Project [SHK+10];
- Deducation STREP Proposal [WSA+12];
- Road to an Intelligent Geometry Book, COST Proposal, OC-2019-1-XXXX.

Intergeo & I2GATP

The I2GATP format is an extension of the I2G (Intergeo) common format aimed to support conjectures and proofs produced by DGSs/GATPs.

XSD files contain the specification of the format:

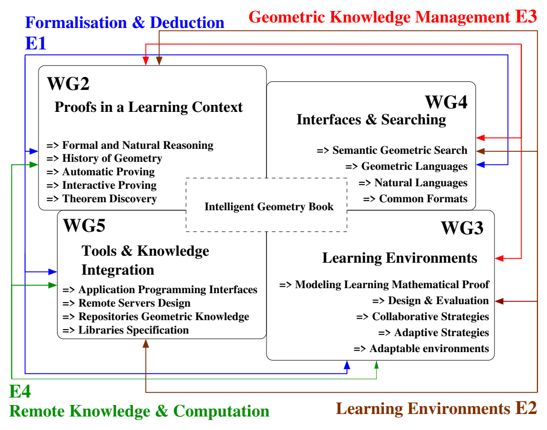
- `information.xsd` with the meta-information about a given geometric problem;
- `intergeo.xsd` no more than the XSD for the I2G format;
- `conjecture.xsd` with the specification of the conjectures;
- `proofInfo.xsd` with the meta-information about the proof(s).

All the XML files containing the information about a geometric problem and also other auxiliary files, are packaged in the I2GATP container, an extension of the I2G container.

A library of programs support the I2GATP format.

*i*GeometryBook

The "Road to an Intelligent Geometry Book" (COST) Action is dedicated to the study of how current developing methodologies and technologies of knowledge representation, management, and discovery in mathematics, can be incorporated effectively into the learning environments of the future.



CISUC

Repositories of Geometric Problems

GeoThms: a Web-based framework for exploring geometrical knowledge that integrates Dynamic Geometry Software (DGS), Automatic Theorem Provers (ATP), and a repository of geometrical constructions, figures and proofs. [JQ06, QJ07].

TGTP: a Web-based library of problems in geometry to support the testing and evaluation of geometric automated theorem proving (GATP) systems [Qua11].

Sets of Examples and Comunities: Intergeo; GeoGebra; Geometriagon; examples in the DGSs/GATPs.)

TGTP

A comprehensive and easily accessible, library of GATP test problems [Qua11].

- Web-based, easily available to the research community. Easy to use.
- Tries to cover the different forms of automated proving in geometry, e.g. synthetic proofs and algebraic proofs.
- provides a mechanism for adding new problems.
- (...)

It is independent of any particular GATP system $\mapsto$ the I2GATP common format [QH12].

Proofs/Readble Proofs/Visual Proofs

Readable Proofs

- What is a readable proofs [QSGB19]?
- Can GATPs produce readable proofs [JNQ12]?

Visual Reasoning

- Proofs with a visual counterpart [QS19].
- Proofs done by "visual means" [YCG10a, YCG10b]

Readability of a Proof

- According to [CGZ94, p.442] a formal proof, done using the area method, is considered readable if one of the following conditions holds:
  - the maximal term in the proof is less than or equal to 5;
  - the number of deduction steps of the proof is less than or equal to 10;
  - the maximal term in the proof is less than or equal to 10 and the deduction step is less than or equal to 20.
- The de Bruijn factor [deB94, Wie00], the quotient of the *size* of corresponding informal proof and the *size* of the formal proof, could also be used as a measure of readability. Using this quotient a proof can be considered readable if the value is less than or equal to 2 (the formal proof is at most twice as larger then a given informal proof).

GATP, Readable Proofs: GCLC Area Method

(1) $\left(\left(\frac{\overrightarrow{AF}}{\overrightarrow{FB}} \cdot \frac{\overrightarrow{BD}}{\overrightarrow{DC}}\right) \cdot \frac{\overrightarrow{CE}}{\overrightarrow{EA}}\right) = 1$, by the statement

(2) $\left(\left(\left(-1 \cdot \frac{\overrightarrow{AF}}{\overrightarrow{BF}}\right) \cdot \frac{\overrightarrow{BD}}{\overrightarrow{DC}}\right) \cdot \frac{\overrightarrow{CE}}{\overrightarrow{EA}}\right) = 1$, by geometric simplifications

(3) $\left(-1 \cdot \left(\frac{\overrightarrow{AF}}{\overrightarrow{BF}} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{\overrightarrow{CE}}{\overrightarrow{EA}}\right)\right)\right) = 1$, by algebraic simplifications

(4) $\left(-1 \cdot \left(\frac{S_{APC}}{S_{BPC}} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{\overrightarrow{CE}}{\overrightarrow{EA}}\right)\right)\right) = 1$, by Lemma 8 (point F eliminated)

(5) $\left(-1 \cdot \left(\frac{S_{APC}}{S_{BPC}} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \left(-1 \cdot \frac{\overrightarrow{CE}}{\overrightarrow{AE}}\right)\right)\right)\right) = 1$, by geometric simplifications

(6) $\frac{\left(S_{APC} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{\overrightarrow{CE}}{\overrightarrow{AE}}\right)\right)}{S_{BPC}} = 1$, by algebraic simplifications

(7) $\frac{\left(S_{APC} \cdot \left(\frac{\overrightarrow{BD}}{\overrightarrow{DC}} \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{S_{BPC}} = 1$, by Lemma 8 (point E eliminated)

(8) $\frac{\left(S_{APC} \cdot \left(\left(-1 \cdot \frac{\overrightarrow{BD}}{\overrightarrow{CD}}\right) \cdot \frac{S_{CPB}}{S_{APB}}\right)\right)}{(-1 \cdot S_{CPB})} = 1$, by geometric simplifications

(9) $\frac{\left(S_{APC} \cdot \frac{\overrightarrow{BD}}{\overrightarrow{CD}}\right)}{S_{APB}} = 1$, by algebraic simplifications

(10) $\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{S_{CPA}}\right)}{S_{APB}} = 1$, by Lemma 8 (point D eliminated)

(11) $\frac{\left(S_{APC} \cdot \frac{S_{BPA}}{(-1 \cdot S_{APC})}\right)}{(-1 \cdot S_{BPA})} = 1$, by geometric simplifications

(12) $1 = 1$, by algebraic simplifications

GATP, Readable Proofs: Coherent Logic

Example: Proof Generated by ArgoCLP

Let us prove that $p = r$ by reductio ad absurdum.

1. Assume that $p \neq r$.
   2. It holds that the point A is incident to the line q or the point A is not incident to the line q (by axiom of excluded middle).
      3. Assume that the point A is incident to the line q.
         4. From the facts that $p \neq q$, and the point A is incident to the line p, and the point A is incident to the line q, it holds that the lines p and q intersect (by axiom ax_D5).
         5. From the facts that the lines p and q intersect, and the lines p and q do not intersect we get a contradiction.

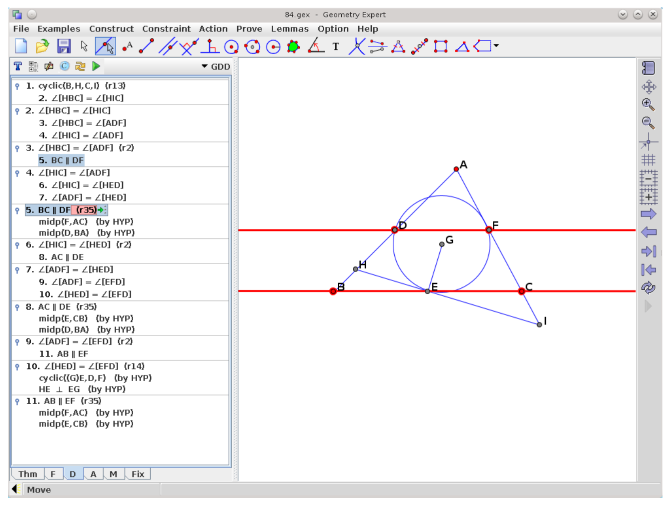
         Contradiction.
      6. Assume that the point A is not incident to the line q.
         7. From the facts that the lines p and q do not intersect, it holds that the lines q and p do not intersect (by axiom ax_nint_l_l_21).
         8. From the facts that the point A is not incident to the line q, and the point A is incident to the plane α, and the line q is incident to the plane α, and the point A is incident to the line p, and the line p is incident to the plane α, and the lines q and p do not intersect, and the point A is incident to the line r, and the line r is incident to the plane α, and the lines q and r do not intersect, it holds that $p = r$ (by axiom ax_E2).
         9. From the facts that $p = r$, and $p \neq r$ we get a contradiction.

         Contradiction.

Therefore, it holds that $p = r$.

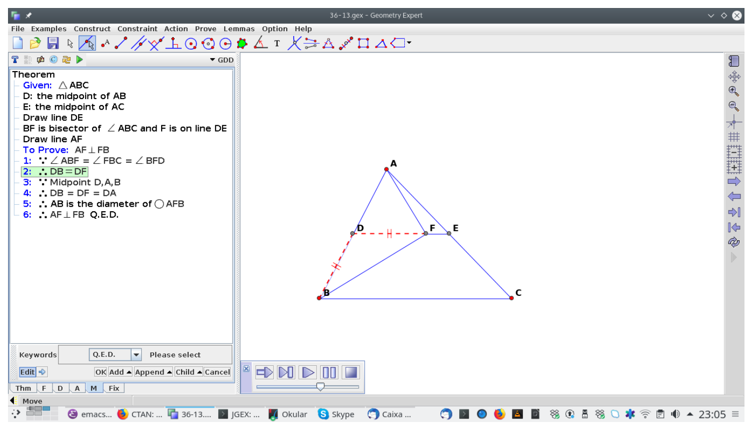
This proves the conjecture.

GATP, Proofs With Visual Support



JGEX – Example 84, Step 2

GATP, Visual Proofs



JGEX – Example 36-13 & PYTH-cnm14

Geometrography

Geometrography, “alias the art of geometric constructions” was proposed by Émile Lemoine between the late 1800s and the early 1900s [SBQ19, Mac93, Lem02, QSGB19].

Measure the complexity of ruler-and-compass geometric constructions.

Coefficient Simplicity: denoting the number of times any particular operation is performed.

Coefficient Exactitude: each time a drawing instrument is used, two types of error can be introduced in the image, systematic error and accidental errors due to personal operator's actions.

Geometrography

Considering the modifications proposed by Mackay [Mac93], the following ruler-and-compass constructions and the corresponding coefficients can be considered.

To place the edge of the ruler in coincidence with one point .... R_1
To place the edge of the ruler in coincidence with two points .. $2R_1$
To draw a straight line .... R_2
To put one point of the compasses on a determinate point ..... C_1
To put one point of the compasses on two determinate points . $2C_1$
To describe a circle .... C_2

For a given construction with $l_1 R_1 + l_2 R_2 + m_1 C_1 + m_2 C_2$ steps.

$cs = l_1 + l_2 + m_1 + m_2$, is called the coefficient of simplicity.
$ce = l_1 + m_1$ is called the coefficient of exactitude.

DGSs & Geometrography

Extrapolating (modernising) geometrography to DGS.

Coefficient of simplicity – must be adapted to new tools.
Coefficient of exactitude – loose its meaning (error free manipulations).
Coefficient of freedom – counts the degrees of freedom, gives a value for the dynamism of the construction.

Geometrography in GCLC (commands in the GCL language): a point in the plane (D), two degrees of freedom; a line defined by two points ($2C$); a point in a line D, one degree of freedom; etc.

Geometrography in GeoGebra: similar to GCLC, but using GeoGebra tools.

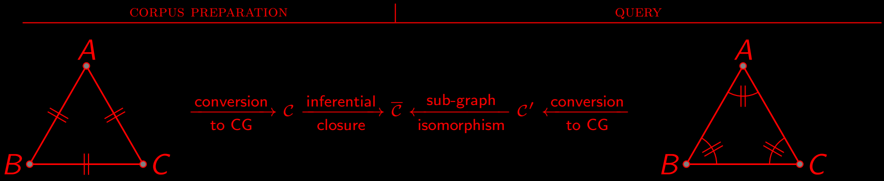
Geometrography as a way to measure the complexity and dynamism of a given construction, being able to compare between different solutions to a same goal

...and how about complexity of a proofs?

Geometric Search

When accessing RGK it should be possible to do geometric searches, i.e. we should be able to provide a geometric construction and look for similar constructions [QH12, HQ14, HQ18] .

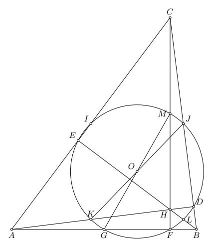
Given (in the RGK) a *triangle with three equal sides*, the query about a *triangle with three equal angles* (which is geometrically equivalent) should be successful.



Taxonomies for Geometry

The usefulness of repositories of geometric knowledge is directly related with the possibility of an easy retrieval of the information a given user is looking for [QSGB19, Qua18].

GEO0316—Nine Points Circle Prove that in any triangle midpoints of each side, feet of each altitude and midpoints of the segments of each altitude from its vertex to the orthocenter lie on a circle [Cho88].



MSC: 51M05, 70G55, 94B27.
GATP Provability: 1/3: *GCLC* area method, "The conjecture is out of scope of the prover"; *GCLC* Wu's method, "The conjecture successfully proved"; *GCLC* Gröbner basis method, "The conjecture not proved - timeout".
Readability [CGZ94]: non-synthetic proof: Wu's Method, 16 pages long proof.
Readability [deB94]: no readable proof: de Bruijn factor: 16/6.
Efficiency (CPU time): 0.17s
CCS: C.A.3; CO.A.1; CO.C.10; CO.D.12.
Construction Complexity: complex (cs=41). $cs = 3\times D + 3\times 2C + 3\times 2C + 3\times 2C + 2\times 2C + 2C + 3\times 2C + 2\times 2C + 2C + 2C = 41$; $cf = 3\times 2 = 6$.
Proofs in Education: Verification: good (0.17s); Explanation: no, only an algebraic, long (16 pages) GATP proof, exist.

MSC—Mathematics Subject Classification (`http://msc2010.org/`)
CCS—Common Core Standard (`http://www.corestandards.org/Math/`)

Conference & Journals

CADE (IJCAR/FLoC) International Conference on Automated Deduction http://www.cadeinc.org/conferences, every year.

ADG International Conference on Automated Deduction in Geometry, http://adg2018.cc4cm.org/ (ADG2018), every two years.

ThEdu Theorem Proving Components for Educational Software http://www.uc.pt/en/congressos/thedu/, workshop at CADE, every year.

JAR Journal of Automated Reasoning, https://link.springer.com/journal/10817, Springer.

LNCS CADE and ADG proceedings, https://www.springer.com/gp/computer-science/lncs.

EPTCS ThEdu post-proceedings, http://www.eptcs.org/.

What to Do Next?

Integration of Methods integrate the study of logical, combinatorial, algebraic, numeric and graphical algorithms with heuristics, knowledge bases and reasoning mechanisms.

Applications design and implement integrated systems for computer geometry, integrating, in a modular fashion, DGSs, ITPs, GATPs, RGPs, etc. in research and/or educational environments.

Higher Geometry The existing algorithms should be extended and improved, new and advanced algorithms be developed to deal with reasoning in different geometric theories.

Axiom Systems Development of new axiom systems, motivated by machine formalisation. [ADM09]

Formalisation formalising geometric theories and methods.

Discovery Automated discovery of new results.

CISUC

Bibliography I

Miguel Abánades, Francisco Botana, Zoltán Kovács, Tomás Recio, and Csilla Sólyom-Gecse.
Development of automatic reasoning tools in geogebra.
ACM Commun. Comput. Algebra, 50(3):85–88, November 2016.

Miguel Á. Abánades, Francisco Botana, Antonio Montes, and Tomás Recio.
An algebraic taxonomy for locus computation in dynamic geometry.
Computer-Aided Design, 56:22 – 33, 2014.

J. R. Anderson, C. F. Boyle, and G. Yost.
The geometry tutor.
The Journal of Mathematical Behavior, pages 5–20, 1986.

Jeremy Avigad, Edward Dean, and John Mumma.
A formal system for Euclid's elements.
The Review of Symbolic Logic,, 2:700–768, 2009.

Francisco Botana and Miguel A. Abánades.
Automatic deduction in dynamic geometry using sage.
In *THedu'11, CTP Components for Educational Software (postproceedings)*, volume 79 of *EPTCS*, 2012.

Francisco Botana, Miguel A. Abánades, and Jesús Escribano.
Computing locus equations for standard dynamic geometry environments.
In Yong Shi, G. Dick van Albada, Jack Dongarra, and Peter M. A. Sloot, editors, *International Conference on Computational Science (2)*, volume 4488 of *Lecture Notes in Computer Science*, pages 227–234. Springer, 2007.

Bibliography II

Pierre Boutry, Gabriel Braun, and Julien Narboux.
From Tarski to Descartes: Formalization of the Arithmetization of Euclidean Geometry.
In James H. Davenport and Fadoua Ghourabi, editors, *SCSS 2016, the 7th International Symposium on Symbolic Computation in Software Science*, volume 39 of *EPiC Series in Computing*, page 15, Tokyo, Japan, March 2016. EasyChair.

Marc Bezem and Thierry Coquand.
Automating coherent logic.
In Geoff Sutcliffe and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning*, volume 3835 of *Lecture Notes in Computer Science*, pages 246–260. Springer Berlin / Heidelberg, 2005.
10.1007/11591191_18.

B. Buchberger, A. Craciun, T. Jebelean, L. Kovacs, T. Kutsia, K. Nakagawa, F. Piroi, N. Popov, J. Robu, M. Rosenkranz, and W. Windsteiger.
Theorema: Towards computer-aided mathematical theory exploration.
Journal of Applied Logic, 4(4):470–504, 2006.

Philippe Balbiani and Luis del Cerro.
Affine geometry of collinearity and conditional term rewriting.
In Hubert Comon and Jean-Pierre Jounnaud, editors, *Term Rewriting*, volume 909 of *Lecture Notes in Computer Science*, pages 196–213. Springer Berlin / Heidelberg, 1995.
10.1007/3-540-59340-3_14.

CISUC

Bibliography III

Francisco Botana, Markus Hohenwarter, Predrag Janičić, Zoltán Kovács, Ivan Petrović, Tomás Recio, and Simon Weitzhofer.
Automated Theorem Proving in GeoGebra: Current Achievements.
Journal of Automated Reasoning, 55(1):39–59, 2015.

Francisco Botana and José L. Valcarce.
A dynamic-symbolic interface for geometric theorem discovery.
Computers and Education, 38:21–35, 2002.

Michael Beeson, Julien Narboux, and Freek Wiedijk.
Proof-checking euclid.
2018.

Francisco Botana.
Interactive versus symbolic approaches to plane loci generation in dynamic geometry environments.
In Peter Sloot, Alfons Hoekstra, C. Tan, and Jack Dongarra, editors, *Computational Science — ICCS 2002*, volume 2330 of *Lecture Notes in Computer Science*, pages 211–218. Springer Berlin / Heidelberg, 2002. 10.1007/3-540-46080-2_22.

B. Buchberger.
Introduction to Gröbner Bases.
In B. Buchberger and F. Winkler, editors, *Gröbner Bases and Applications*, number 251 in London Mathematical Society Lecture Notes Series, pages 3–31. Cambridge University Press, 1998.

Bibliography IV

Giuseppa Carrá Ferro, Giovanni Gallo, and Rosario Gennaro.
Probabilistic verification of elementary geometry statements.
In Dongming Wang, editor, *Automated Deduction in Geometry*, volume 1360 of *Lecture Notes in Computer Science*, pages 87–101. Springer Berlin / Heidelberg, 1997.
10.1007/BFb0022721.

Pedro Cobo, Josep Fortuny, Eloi Puertas, and Philippe Richard.
AgentGeom: a multiagent system for pedagogical support in geometric proof problems.
International Journal of Computers for Mathematical Learning, 12:57–79, 2007.
10.1007/s10758-007-9111-5.

Shang-Ching Chou and Xiao-Shan Gao.
Ritt-wu's decomposition algorithm and geometry theorem proving.
In Mark E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *LNCS*, pages 207–220, Berlin, Heidelberg, 1990. Springer Berlin Heidelberg.

Shang-Ching Chou and Xiao-Shan Gao.
Automated reasoning in geometry.
In John Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, pages 707–749. Elsevier Science Publishers B.V., 2001.

Shang-Ching Chou, Xiao-Shan Gao, and Zheng Ye.
Java geometry expert.
`http://www.cs.wichita.edu/~ye/`, 2004.

Bibliography V

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
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*, pages 48–56. IEEE Computer Society Press, June 1993.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
Machine Proofs in Geometry.
World Scientific, 1994.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
Automated production of traditional proofs in solid geometry.
Journal of Automated Reasoning, 14:257–291, 1995.

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

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
Automated generation of readable proofs with geometric invariants, II. theorem proving with full-angles.
Journal of Automated Reasoning, 17(13):349–370, 1996.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.
A deductive database approach to automated geometry theorem proving and discovering.
Journal of Automated Reasoning, 25:219–246, 2000.

Bibliography VI

S.C. Chou.
Proving and discovering geometry theorems using Wu's method.
PhD thesis, The University of Texas, Austin, 1985.

Shang-Ching Chou.
An Introduction to Wu's Method for Mechanical Theorem Proving in Geometry.
Journal of Automated Reasoning, 4:237–267, 1988.

H. Coelho and L. M. Pereira.
Geom: A Prolog geometry theorem prover.
Memórias 525, Laboratório Nacional de Engenharia Civil, Ministério de Habitação e Obras Públicas, Portugal, 1979.

H. Coelho and L. M. Pereira.
Automated reasoning in geometry theorem proving with Prolog.
Journal of Automated Reasoning, 2(4):329–390, 1986.

Christophe Dehlinger, Jean-François Dufourd, and Pascal Schreck.
Higher-order intuitionistic formalization and proofs in Hilbert's elementary geometry.
In Dongming Wang Jürgen Richter-Gebert, editor, *Proceedings of Automated Deduction in Geometry (ADG00)*, volume 2061 of *Lecture Notes in Computer Science*, pages 306–324, 2000.

N. G. deBruijn.
Selected Papers on Automath, volume 133 of *Studies in logic and the foundations of mathematics*, chapter A survey of the project Automath, pages 41–161.
North-Holland, Amsterdam, 1994.

Bibliography VII

Jesús Escribano, Francisco Botana, and Miguel A. Abánades.
Adding remote computational capabilities to dynamic geometry systems.
Mathematics and Computers in Simulation, 80(6):1177 – 1184, 2010.

Laurent Fuchs and Laurent Théry.
A formalization of grassmann-cayley algebra in coq and its application to theorem proving in projective geometry.
In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 51–67. Springer Berlin Heidelberg, 2011.

H. Gelernter.
Realization of a geometry-theorem proving machine.
In *Computers & thought*, pages 134–152, Cambridge, MA, USA, 1959. MIT Press.

H. Gelernter, J. R. Hansen, and D. W. Loveland.
Empirical explorations of the geometry theorem machine.
In *Papers presented at the May 3-5, 1960, western joint IRE-AIEE-ACM computer conference*, IRE-AIEE-ACM '60 (Western), pages 143–149, New York, NY, USA, 1960. ACM.

Paul C. Gilmore.
An examination of the geometry theorem machine.
Artif. Intell., 1(3):171–187, 1970.

René Grothmann.
About C.a.R.
`http://compute.ku-eichstaett.de/MGF/wikis/caruser/doku.php?id=history`, 2011.

Bibliography VIII

Frédérique Guilhot.
Formalisation en Coq d'un cours de géométrie pour le lycée.
In *Journées Francophones des Langages Applicatifs*, Janvier 2004.

David Hilbert.
Foundations of Geometry.
Open Court Publishing, 1977.
10th Revised edition. Editor: Paul Barnays.

M. Hadzikadic, F. Lichtenberger, and D. Y. Y. Yun.
An application of knowledge-base technology in education: a geometry theorem prover.
In *Proceedings of the fifth ACM symposium on Symbolic and algebraic computation*, SYMSAC '86, pages 141–147, New York, NY, USA, 1986. ACM.

M Hohenwarter.
Geogebra - a software system for dynamic geometry and algebra in the plane.
Master's thesis, University of Salzburg, Austria, 2002.

Yannis Haralambous and Pedro Quaresma.
Querying geometric figures using a controlled language, ontological graphs and dependency lattices.
In S. Watt et al., editor, *CICM 2014*, volume 8543 of *LNAI*, pages 298–311. Springer, 2014.

Yannis Haralambous and Pedro Quaresma.
Geometric search in TGTP.
In Hongbo Li, editor, *Proceedings of the 12th International Conference on Automated Deduction in Geometry*. SMS International, 2018.

CISUC

Bibliography IX

N Jackiw.
The Geometer's Sketchpad v4.0.
Key Curriculum Press, 2001.

Predrag Janičić.
GCLC — A tool for constructive euclidean geometry and more than that.
In Andrés Iglesias and Nobuki Takayama, editors, *Mathematical Software - ICMS 2006*, volume 4151 of *Lecture Notes in Computer Science*, pages 58–73. Springer, 2006.

Predrag Janičić, Julien Narboux, and Pedro Quaresma.
The Area Method: a recapitulation.
Journal of Automated Reasoning, 48(4):489–532, 2012.

Predrag Janičić and Pedro Quaresma.
System description: GCLCprover + GeoThms.
In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 145–150. Springer, 2006.

Predrag Janičić and Pedro Quaresma.
Automatic verification of regular constructions in dynamic geometry systems.
In Francisco Botana and Tomás Recio, editors, *Automated Deduction in Geometry*, volume 4869 of *Lecture Notes in Computer Science*, pages 39–51. Springer, 2007.

Kenneth R. Koedinger and John R. Anderson.
Abstract planning and perceptual chunks: Elements of expertise in geometry.
Cognitive Science, 14(4):511–550, 1990.

Bibliography X

Gilles Kahn.
Constructive geometry according to Jan von Plato.
Coq contribution, 1995.

Deepak Kapur.
Using Gröbner bases to reason about geometry problems.
Journal of Symbolic Computation, 2(4):399–408, 1986.

Michelle Y. Kim.
Visual reasoning in geometry theorem proving.
In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume II of *IJCAI-89*, pages 1617–1622, Detroit, 1989.

Zoltán Kovács.
The relation tool in geogebra 5.
In Francisco Botana and Pedro Quaresma, editors, *Automated Deduction in Geometry*, pages 53–71. Springer International Publishing, 2015.

Émile Lemoine.
Géométrographie ou Art des constructions géométriques, volume 18 of *Phys-Mathématique*. Scentia, 1902.

J. M. Laborde and R. Strässer.
Cabri-géomètre: A microworld of geometry guided discovery learning.
International reviews on mathematical education- Zentralblatt fuer didaktik der mathematik, 90(5):171–177, 1990.

Bibliography XI

J. S. Mackay.
The geometrography of euclid's problems.
Proceedings of the Edinburgh Mathematical Society, 12:2–16, 1893.

Laura Meikle and Jacques Fleuriot.
Formalizing Hilbert's Grundlagen in Isabelle/Isar.
In David A. Basin and Burkhart Wolff, editors, *Theorem Proving in Higher Order Logics*, volume 2758 of *Lecture Notes in Computer Science*, pages 319–334. Springer, 2003.

Nicolas Magaud, Julien Narboux, and Pascal Schreck.
Formalizing Projective Plane Geometry in Coq.
In Thomas Sturm and Christoph Zengler, editors, *Automated Deduction in Geometry*, volume 6301 of *Lecture Notes in Artifical Intelligence*, pages 141–162. Springer, 2011.

Filip Marić, Ivan Petrović, Danijela Petrović, and Predrag Janičić.
Formalization and implementation of algebraic methods in geometry.
In *First Workshop on CTP Components for Educational Software (THedu'11)*, volume 79 of *EPTCS*, 2012.

Thiago Guimaraes Moraes, Flávia Maria Santoro, and Marcos R.S. Borges.
Tabulæ: educational groupware for learning geometry.
In *Advanced Learning Technologies, 2005. ICALT 2005. Fifth IEEE International Conference on*, pages 750 – 754, july 2005.

R Moriyón, F Saiz, and M Mora.
GeoThink: An Environment for Guided Collaborative Learning of Geometry, volume 4 of *Nuevas Ideas en Informática Educativa*, pages 200–2008.
J. Sánchez (ed), Santiago de Chile, 2008.

Bibliography XII

Noboru Matsuda and Kurt VanLehn.
Advanced geometry tutor: An intelligent tutor that teaches proof-writing with construction.
In Chee-Kit Looi, Gordon I. McCalla, Bert Bredeweg, and Joost Breuker, editors, *Artificial Intelligence in Education - Supporting Learning through Intelligent and Socially Informed Technology, Proceedings of the 12th International Conference on Artificial Intelligence in Education, AIED 2005, July 18-22, 2005, Amsterdam, The Netherlands*, volume 125 of *Frontiers in Artificial Intelligence and Applications*, pages 443–450. IOS Press, 2005.

Julien Narboux.
Formalisation et Automatisation du Raisonnement Géométrique en Coq.
PhD thesis, Université de Paris Sud, 2006.

Julien Narboux.
A graphical user interface for formal proofs in geometry.
Journal of Automated Reasoning, 39:161–180, 2007.

Julien Narboux.
Mechanical theorem proving in Tarski's geometry.
In *Proceedings of Automatic Deduction in Geometry 06*, volume 4869 of *Lecture Notes in Artificial Intelligence*, pages 139–156. Springer-Verlag, 2007.

Julien Narboux.
Formalization of the area method.
Coq user contribution, 2009.
http://dpt-info.u-strasbg.fr/~narboux/area_method.html.

Bibliography XIII

Arthur J. Nevins.
Plane geometry theorem proving using forward chaining.
AI Lab memo 303, MIT, Jan 1974.

Lawrence C. Paulson.
Isabelle: A Generic Theorem Prover, volume 828 of *LNCS*.
Springer-Verlag, 1994.

Lawrence C. Paulson and Tobias Nipkow.
Isabelle tutorial and user's manual.
Technical Report 189, University of Cambridge, Computer Laboratory, January 1990.

Pedro Quaresma and Yannis Haralambous.
Geometry Constructions Recognition by the Use of Semantic Graphs.
In Nuno Cid Martins, Verónica Vasconcelos, Fernando Lopes, Inácio Fonseca, Jorge Barbosa, Nuno Rodrigues, and Simão Paredes, editors, *Atas da XVIII Conferência Portuguesa de Reconhecimento de Padrões (RecPad 2012)*, pages 47–48. Instituto Superior de Engenharia de Coimbra, Tipografia Damasceno, October 2012.
ISBN: 978-989-8331-15-1.

Pedro Quaresma and Predrag Janičić.
Framework for constructive geometry (based on the area method).
Technical Report 2006/001, Centre for Informatics and Systems of the University of Coimbra, 2006.

Bibliography XIV

Pedro Quaresma and Predrag Janičić.
Integrating dynamic geometry software, deduction systems, and theorem repositories.
In Jonathan M. Borwein and William M. Farmer, editors, *Mathematical Knowledge Management*, volume 4108 of *Lecture Notes in Computer Science*, pages 280–294, Berlin, 2006. Springer.

Pedro Quaresma and Predrag Janičić.
GeoThms – a Web System for euclidean constructive geometry.
Electronic Notes in Theoretical Computer Science, 174(2):35 – 48, 2007.

Pedro Quaresma and Vanda Santos.
Proof Technology in Mathematics Research and Teaching, chapter Computer-generated geometry proofs in a learning context.
Springer, 2019.
(in press).

Pedro Quaresma, Vanda Santos, Pierluigi Graziani, and Nuno Baeta.
Taxonomies of geometric problems.
Journal of Symbolic Computation, (in press), 2019.

Pedro Quaresma, Vanda Santos, and Milena Marić.
WGL, a web laboratory for geometry.
Education and Information Technologies, 23(1):237–252, Jan 2018.

Art Quaife.
Automated development of Tarski's geometry.
Journal of Automated Reasoning, 5:97–118, 1989.
10.1007/BF00245024.

Bibliography XV

Pedro Quaresma.
Thousands of Geometric problems for geometric Theorem Provers (TGTP).
In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 169–181. Springer, 2011.

Pedro Quaresma.
Automatic deduction in an AI geometry book.
In Jacques Fleuriot, Dongming Wang, and Jacques Calmet, editors, *Artificial Intelligence and Symbolic Computation*, volume 11110 of *Lecture Notes in Artificial Intelligence*, pages 221–226. Springer International Publishing, 2018.

Philippe R. Richard, Josep M Fortuny, Markus Hohenwarter, and Michel Gagnon.
geogebraTUTOR: une nouvelle approche pour la recherche sur l'apprentissage compétentiel et instrumenté de la géométrie à l'école secondaire.
In Theo Bastiaens and Saul Carliner, editors, *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education 2007*, pages 428–435, Quebec City, Canada, October 2007. Association for the Advancement of Computing in Education (AACE).

Jürgen Richter-Gebert and Ulrich Kortenkamp.
The Interactive Geometry Software Cinderella.
Springer, 1999.

Judit Robu.
Geometry Theorem Proving in the Frame of the Theorema Project.
PhD thesis, Johannes Kepler Universität, Linz, September 2002.

Bibliography XVI

Judit Robu.
Automated Proof of Geometry Theorems Involving Order Relation in the Frame of the Theorema Project.
In Horia F. Pop, editor, *Knowledge Engineering: Principles and Techniques*, number Special Issue in Studia Universitatis "Babes-Bolyai", Series Informatica, pages 307–315, 2007.

Vanda Santos, Nuno Baeta, and Pedro Quaresma.
Geometrography in dynamic geometry.
International Journal for Technology in Mathematics Education, 2019.
accepted.

E. Santiago, Maxim Hendriks, Yves Kreis, Ulrich Kortenkamp, and Daniel Marquès.
I2G Common File Format Final Version.
Technical Report D3.10, The Intergeo Consortium, 2010.

Sana Stojanović, Vesna Pavlović, and Predrag Janičić.
A coherent logic based geometry theorem prover capable of producing formal and readable proofs.
In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 201–220. Springer Berlin Heidelberg, 2011.

Vanda Santos and Pedro Quaresma.
eLearning course for Euclidean Geometry.
In *Proceedings of the 8th IEEE International Conference on Advanced Learning Technologies, July 1st- July 5th, 2008, Santander, Cantabria, Spain*, pages 387–388, 2008.

Bibliography XVII

Vanda Santos and Pedro Quaresma.
Adaptative Learning Environment for Geometry, volume Advances in Learning Processes, chapter 5, pages 71–92.
I-Tech Education and Publishing KG, Vienna, Austria, 2010.

Vanda Santos, Pedro Quaresma, Milena Marić, and Helena Campos.
Web geometry laboratory: case studies in portugal and serbia.
Interactive Learning Environments, 26(1):3–21, 2018.

The Coq Development Team.
The Coq Proof Assistant, Reference Manual, Version 8.2.
TypiCal Project, Lyon, France, 2009.

Jan von Plato.
The axioms of constructive geometry.
In *Annals of Pure and Applied Logic*, volume 76, pages 169–200, 1995.

Dongming Wang.
Geometry machines: From ai to smc.
In Jacques Calmet, John Campbell, and Jochen Pfalzgraf, editors, *Artificial Intelligence and Symbolic Mathematical Computation*, volume 1138 of *Lecture Notes in Computer Science*, pages 213–239. Springer Berlin / Heidelberg, 1996.
10.1007/3-540-61732-9_60.

Bibliography XVIII

Freek Wiedijk.
The de Bruijn factor.
Poster at International Conference on Theorem Proving in Higher Order Logics (TPHOL2000), 2000. Portland, Oregon, USA, 14-18 August 2000.

Burkhart Wolff, Pascal Schreck, Serge Autexier, Achim D. Brucker, Wolfgang Schreiner, Pedro Quaresma, Ralph-Johan Back, Predrag Janicic, Christian Gütl, and Markus Hohenwarter.
Deductive framework for math-oriented collaborative teaching environments.
Small or Medium-scale focused Research Project (STREP) ICT Call 8: FP7-ICT-2011-8 proposal, January 2012.

Wu Wen-Tsun.
Basic principles of mechanical theorem proving in elementary geometries.
Journal of Automated Reasoning, 2:221–252, 1986.
10.1007/BF02328447.

W.-T. Wu.
The characteristic set method and its application.
In X.-S. Gao and D. Wang, editors, *Mathematics Mechanization and Applications*, pages 3–41, San Diego, CA, 2000. Academic Press.

Zheng Ye, Shang-Ching Chou, and Xiao-Shan Gao.
Visually dynamic presentation of proofs in plane geometry, part 1.
J. Autom. Reason., 45:213–241, October 2010.

Bibliography XIX

Zheng Ye, Shang-Ching Chou, and Xiao-Shan Gao.
Visually dynamic presentation of proofs in plane geometry, part 2.
J. Autom. Reason., 45:243–266, October 2010.