Automating Geometry Construction Problems

Vesna Marinković

Faculty of Mathematics, University of Belgrade, Serbia

ADG 2025 & ThEdu'25, Stuttgart

Table of Contents

ArgoTriCS

Table of Contents

Geometry constraint solving

- Geometric constraint solving involves determining the positions of geometric elements (points, lines, circles) that meet a specified set of constraints
- Constraints considered include distances, incidences, tangencies, etc.
- One of central tasks is reasoning about constructibility – ability to achieve a given configuration with specific tools
- Geometric construction solving is a special case of geometric constraint solving

Geometry construction problems

- One of the most studied (and most challenging) problems in math education
- Require visualization, precision, abstract way of thinking
- The task is to describe a construction of a geometric figure that meets given constraints
- Constructions are procedures

Phases in solving construction problems

- Analysis: analyzing the figure to be constructed and finding key properties for construction
- Construction: identifying the sequence of steps used to construct the figure
- Proof: proving that the constructed figure satisfies the problem specification
- Discussion: analyzing when and how many solutions exist

Constructions using straightedge and compass

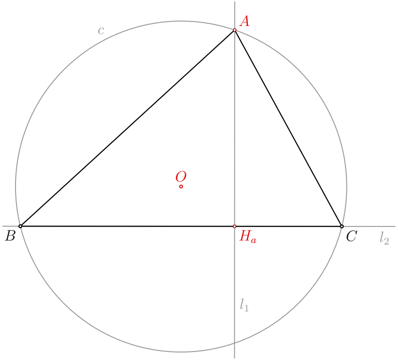
- Tools: straightedge and compass
- Elementary steps:
  - construction of an arbitrary point
  - construction of a line through two given points
  - construction of a circle centered at given point passing through another point
  - construction of an intersection of two circles, two lines, or a line and a circle
- We usually use also compound construction steps

Location triangle construction problems

- Construction problems may use locational data (positions) or non-locational data (lengths, angles, ratios)
- The first type of problems is known as location construction problems
- Triangle construction problems require constructing a triangle that meets given conditions
- Triangle construction problems are easy to state, but challenging to solve
- They play important role in both educational context and real-world applications
- In this talk we focus on triangle location construction problems

Motivating example

- Example: Construct a triangle ABC given locations of
  - its vertex A
  - its altitude foot H_a
  - its circumcenter O
- Solution: A, H_a, O



1. Construct the line l_1 through points A and H_a
2. Construct the perpendicular l_2 to line l_1 through H_a
3. Construct the circle c centered at O passing through A
4. Construct points B and C as the intersection points of l_2 and c

Table of Contents

Towards automating constructions

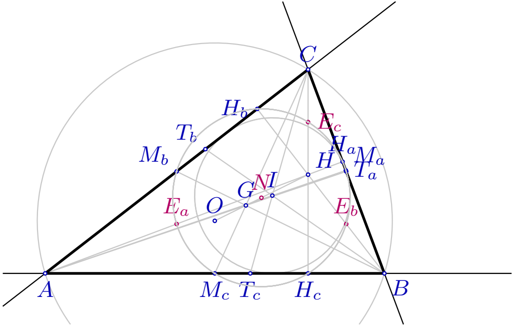
- The main problem in solving:
  combinatorial explosion – huge search space
  - many different construction steps available
  - plenty of objects that each step could be applied to
- Some instances are unsolvable (e.g. angle trisection)
- Having automated tool for construction solving is important for many reasons
- However, there are only a few tools for automating constructions (Schreck; Chou, Gao, Zhang; Gulwani)

ArgoTriCS

- ArgoTriCS – Argo Triangle Construction Solver
- System for automated solving of location construction problems from the given corpus
- Primarely focused on triangle construction problems:
  constructing $\triangle ABC$ if locations of three significant points in the triangle are given
- It can be applied to other types of construction problems, also
- Developed in Prolog
- Based on systematization of background geometrical knowledge needed

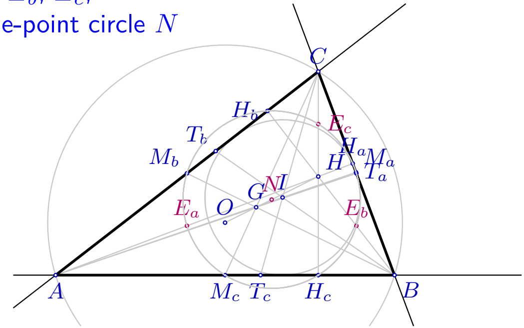
Corpora of triangle location construction problems

- Wernick's corpus (1982)
  - vertices A, B, C,
  - side midpoints M_a, M_b, M_c,
  - feet of altitudes H_a, H_b, H_c,
  - feet of internal angle bisectors T_a, T_b, T_c.
  - orthocenter H, centroid G, circumcenter O, incenter I,



Corpora of triangle location construction problems

- Wernick's corpus (1982)
- Connelly's corpus, extension of Wernick's corpus (2009)
  - vertices A, B, C,
  - side midpoints M_a, M_b, M_c,
  - feet of altitudes H_a, H_b, H_c,
  - feet of internal angle bisectors T_a, T_b, T_c,
  - orthocenter H, centroid G, circumcenter O, incenter I,
  - Euler points E_a, E_b, E_c,
  - the center of nine-point circle N



Possible statuses of construction problems

- Construction problems can be
  - solvable – S (A, B, M_a)
  - unsovable – U (T_a, T_b, T_c)
  - redundant – R (A, B, M_c)
  - locus-dependent – L (A, B, O)

Algorithm for solving geometric construction problems

- Search starts from the objects given
- Search stops once all vertices are constructed or there are no more applicable primitive constructions
- Iterative procedure with forward chaining
- Primitive constructions are applied in a waterfall manner
- Objects that can be constructed are only relevant ones
- Obtained proof-trace is filtered-out and only relevant steps are preserved in the clean proof

Solution to geometric construction problem

- Informal description of construction in NL form
- Formal specification of construction in GCLC language and JSON format
- Static ilustration in GCLC tool with step-by-step animation
- Dynamic illustration using ArgoDG library
- Construction correctness proof obtained by
  - OpenGeoProver (Wu's method)
  - Provers within GCLC tool (area method, Gröbner basis method and Wu's method)
- Non-degeneracy conditions under which the solution exists
- Determination conditions under which the solution is uniquely determined

Example construction in NL form and in GCLC

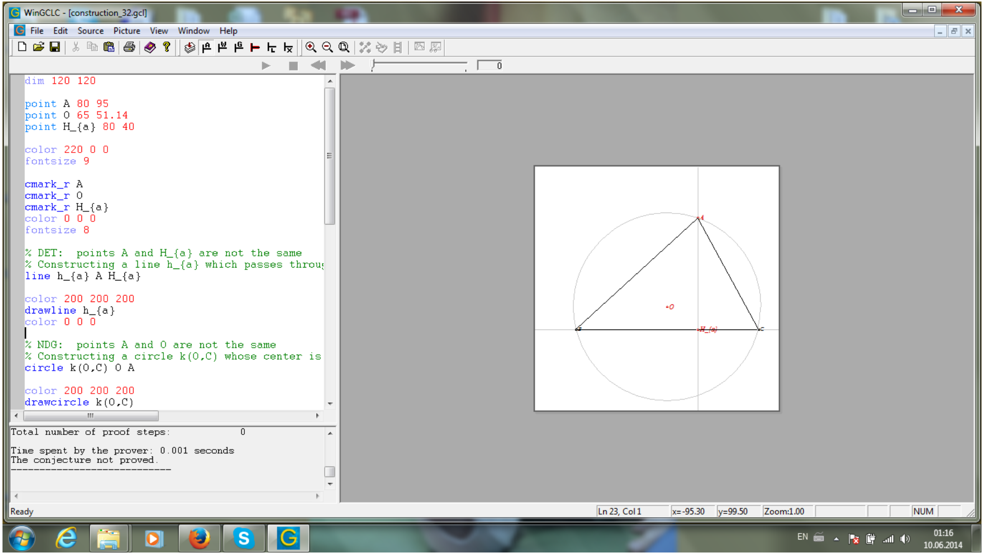
Problem: Given points A, O, and H_a construct the triangle ABC.

Construction:

1. Construct the line h_a through points A and H_a;
2. Construct the perpendicular a to line h_a through point H_a;
3. Construct the circle $\mathcal{C}_c$ with center at point O passing through point A;
4. Construct the intersection points B and C of circle $\mathcal{C}_c$ and line a.

NDG conditions: line a and circle $\mathcal{C}_c$ intersect; points A and O are not the same.
Determination conditions: points A and H_a are not the same.

Example construction in GCLC and corresponding illustration



Evaluation of construction generation using ArgoTriCS

- Solved 66 out of 74 solvable problems from Wernick's corpus and 62 out of 74 from Connelly's corpus
- Detected all redundant and locus-dependent construction problems
- Longest generated construction has 19 construction steps
- For some solvable problems that ArgoTriCS failed to solve, there are no synthetic solutions in the literature
- Automatically generated compendium of solutions to problems from Wernick's and Connely's corpora available online
  http://poincare.matf.bg.ac.rs/~vesnap/animations/compendiums.html

References

MJ2012 V. Marinković, P. Janičić,
Towards Understanding Triangle Construction Problems,
Conference on Intelligent Computer Mathematics (CICM) 2012,
vol. 7362 of *Lecture Notes in Artificial Inteligence*, pp. 126–141, 2012.

M2015 V. Marinković,
On-line Compendium of Triangle Construction Problems with
Automatically Generated Solutions,
The Teaching of Mathematics, XVIII_1, pp. 29–44, 2015.

M2017 V. Marinković,
ArgoTriCS - Automated Triangle Construction Solver,
Journal of Experimental & Theoretical Artificial Intelligence,
vol. 29, no. 2, pp. 247–271, 2017.

M2023 V. Marinković:
Prolog in Automated Reasoning in Geometry,
in book: *Prolog: The Next 50 Years*, *Lecture Notes in Artificial Intelligence*,
vol. 13900, pp. 334–345, 2023.

Table of Contents

Proving unconstructibility

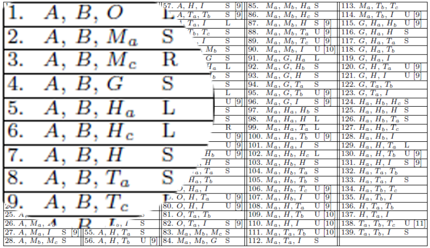
- So far, we were focused solely on solving construction problems
- In schools, also, an emphasize is usually put on solvable construction problems
- How do we know if a construction problem is solvable?
- What about unsolvable construction problems?
- Proofs of RC-unconstructibility are usually performed using algebraic methods
- Even with modern computer algebra systems, it remains challenging

Computer assisted resolving of status of problems

- We focused on determining the status of problems from Wernick's and Connelly's corpora
- In the original Wernick's list, 41 problems were with unknown status, while two problems were incorrectly classified
- In the meanwhile only 15 problems from Wernick's list remained with unknown status
- Automatic tool in Maple, based on algebraization of geometric constructions and Galois theory, resolved status of all problems in Wernick's list (P. Schreck, P. Mathis)
- The status of all problems from Connelly's corpus was also determined, by using additional preprocessing for some problems

Statistics of Wernick's and Connely's corpora

- Wernick's corpus: in total $\binom{16}{3} = 560$ instances, 139 non-trivial, significantly different problems; 3 redundant (R); 23 locus dependent (L); 74 solvable (S); 39 unsolvable (U)



- Connelly's corpus: 580 new instances, 140 non-trivial, significantly different problems; 5 redundant (R); 19 locus dependent (L); 74 solvable (S); 42 unsolvable (U)

Proving (un)constructibility by reduction

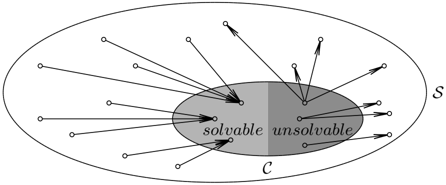
- Determining the status of one problem can be reduced to determining the status of another
- Let us concentrate on triangle construction problems from Wernick's corpus
- There is a reduction from triangle construction problem $P : (A_1, A_2, A_3)$ to problem $P' : (A'_1, A'_2, A'_3)$ if each point from P' can be constructed using the points from P
- It holds:
  - if P' is **S**, then P is also **S**; if P is **U**, then P' is either **U** or **R** or **L**
  - if P is **L**, then P' is **R** or **L**; if P is **R**, then P' is also **R**

Production rules

- How to effectively find reductions between triangle construction problems?
- If, given a set of significant points S, a significant point X is constructible, we say that there is a production rule $S \to X$
- For instance: $\{A, B\} \to M_c$, $\{A, H, T_a, I\} \to B$
- In this way the reduction phase boils down to a simple syntactical procedure
- Production rules can be derived using ArgoTriCS, starting from $k = 2, 3, 4$ points and trying to construct remaining $16 - k$ significant points

Constructibility classes of construction problems

- The goal is to find minimal subset $\mathcal{C}$ of the set of construction problems $\mathcal{S}$ that satisfies:
  - for each problem $P \in \mathcal{S}$ that is **S**, there is a problem $P' \in \mathcal{C}$ such that P' can be obtained from P by application of production rules
  - for each problem $P \in \mathcal{S}$ that is **U**, there is a problem $P' \in \mathcal{C}$ such that P can be obtained from P' by application of production rules
- The problems from $\mathcal{C}$ are key problems
- The problem reduces to a set covering problem, solvable by SAT-based system URSA
- URSA returned the set $\mathcal{C}$ consisting of 9 **S** problems and 33 **U** problems



References

SMMJ2016 P. Schreck, P. Mathis, V. Marinković, P. Janičić,
Wernick's List: A Final Update,
Forum Geometricorum, vol. 16, pp. 69–80, 2016.

SMJ2016 P. Schreck, V. Marinković, P. Janičić,
Constructibility Classes for Triangle Location Problems,
Mathematics in Computer Science, vol. 10, no. 1, pp. 27–39, 2016.

Table of Contents

Solving construction problem as proving a theorem

- Solving construction problems can be approached from the logical point of view
- It corresponds to proving constructively a theorem of the form:

$$\forall X . \exists Y . \Psi(X, Y)$$

- Witness for Y represents a construction of a solution and must involve only points that are constructible from X
- Some construction problems have solutions only under some additional assumptions
- The complete characterization of solvability requires proving the following theorem:

$$\forall X . (\Phi(X) \Leftrightarrow \exists Y . \Psi(X, Y))$$

- $\Phi(X)$ denote contraints on given objects X, not known in advance

Solving construction problems as proving theorems

- Problem of constructing $\triangle ABC$ given its vertex A, altitude foot H_a, and circumcenter O is translated into the following theorem:

$$\begin{aligned}\forall A, H_a, O.(\Phi(A, H_a, O) \quad &\Leftrightarrow \quad \exists B, C.(\neg collinear(A, B, C) \\ &\wedge \quad altitude\_foot(H_a, A, B, C) \wedge circumcenter(O, A, B, C)))\end{aligned}$$

- $\Phi(A, H_a, O)$ has to be discovered, while $\neg collinear(A, B, C)$ is implicit goal condition

Machine-verifiable solutions to construction problems

- Generation of formal solutions to construction problems requires synergy of tools:
  - tool for solving constructions – used for construction phase
  - algebraic theorem provers – used for proving phase
  - synthetic automated theorem provers – used for analysis and proof phase
  - interactive theorem provers – used for gluing obtained proof fragments
- Formal proofs for a subset of problems from Wernick's corpus are obtained
- Two gaps:
  - link to external algebraic provers
  - using higher-order lemmas as axioms

References

MJS2014 V. Marinković, P. Janičić, P. Schreck,
Solving Geometric Construction Problems Supported by Theorem Proving,
Automated Deduction in Geometry (ADG) 2014.

MJS2015 V. Marinković, P. Janičić, P. Schreck,
Computer Theorem Proving for Verifiable Solving of Geometric Construction Problems,
Automated Deduction in Geometry (ADG) 2014,
vol. 9201 of *Lecture Notes in Computer Science*, pp. 72–93, 2015.

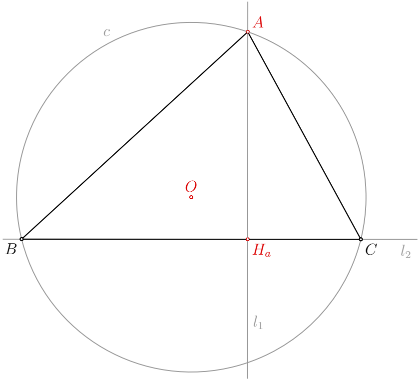
Table of Contents

Proving constructions correct

- Most of existing systems for automated solving geometric constructions do not consider proving generated constructions correct
- Proving triangle construction correct means proving that the points given by problem setting are indeed the corresponding points of the constructed triangle

Motivating example – construction phase

- Task: Construct $\triangle ABC$ given its vertex A, altitude foot H_a, and circumcenter O

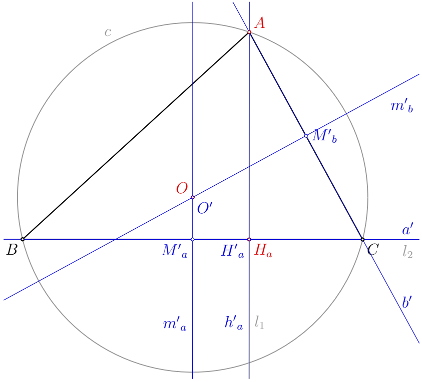


1. Construct the line l_1 through points A and H_a
2. Construct the perpendicular l_2 to line l_1 through H_a
3. Construct the circle c centered at O containing A
4. Construct points B and C as the intersection points of l_2 and c

- Solution: A, H_a, O

Motivating example – correctness conjecture

- ArgoTriCS generates correctness conjectures that can be passed to automated provers
- We should prove that for constructed $\triangle ABC$, A is its vertex, H_a is its altitude foot and O is circumcenter



1. Construct the line b' through points A and C
2. Construct the midpoint M_b' of the segment AC
3. Construct the line a' through points B and C
4. Construct the midpoint M_a' of the segment BC
5. Construct the perpendicular m_a' to a' through M_a
6. Construct the perpendicular m_b' to b' through M_b
7. Construct the intersection point O' of m_a' and m_b'
8. Construct the perpendicular h_a' to a' through A
9. Construct the intersection point H_a' of a' and h_a'
10. Prove that O is identical to O'
11. Prove that H_a is identical to H_a'

Algebraic and semi-synthetic proofs

- Algebraic methods (Wu's method, Gröbner basis method) – very efficient, but non-readable
- Semi-synthetic methods (area method) – readable proofs, but formulated in non-traditional terms

5.9 Triangulation, step 9

Choosing variable: Trying the variable with index 4.

Variable x_4 selected: The number of polynomials with this variable is 2.

Minimal degrees: 2 polynomials with degree 1 and 3 polynomials with degree 2.

Polynomial with linear degree: Removing variable x_4 from all other polynomials by reducing them with polynomial p_3.

Finished a triangulation step, the current system is:

$$
\begin{aligned}
p_0 &= u_2x_2 + u_3x_1 + (-u_3^2 - u_2^2)\\
p_1 &= x_2^2 - 2u_1x_2 + x_1^2\\
p_2 &= x_3^2x_2^2 - 2u_2x_3^2x_2 + x_3^2x_1^2 - 2u_3x_3^2x_1 +\\
&\quad (u_3^2 + u_2^2)x_3^2 - 2u_3x_3x_2^2 + (2u_2 - 2u_1)x_3x_2x_1 + (2u_3u_2 + 2u_3u_1)x_3x_2 +\\
&\quad (-2u_2^2 + 2u_2u_1)x_3x_1 - 2u_3u_2u_1x_3 + u_3^2x_2^2 +\\
&\quad (-2u_3u_2 + 2u_3u_1)x_2x_1 - 2u_3^2u_1x_2 + (u_2^2 - 2u_2u_1)x_1^2 + 2u_3u_2u_1x_1
\end{aligned}
$$

$$
\begin{aligned}
p_4 &= -x_6x_1 + x_5x_2\\
p_5 &= -x_6 + 0.5x_2\\
p_6 &= -x_8x_3 + x_8x_1 + x_7x_4 - x_7x_2 - x_4x_1 + x_3x_2\\
p_7 &= -x_8 + 0.5x_4 + 0.5x_2\\
p_8 &= -x_{10}x_4 + x_{10}x_2 - x_9x_3 + x_9x_1 + x_8x_4 - x_8x_2 + x_7x_3 - x_7x_1\\
p_9 &= x_{10}x_2 + x_9x_1 - x_6x_2 - x_5x_1\\
p_{10} &= -x_{12}x_4 + x_{12}x_2 - x_{11}x_3 + x_{11}x_1\\
p_{11} &= x_{12}x_3 - x_{12}x_1 - x_{11}x_4 + x_{11}x_2 + x_4x_1 - x_3x_2
\end{aligned}
$$

1. Creating S-polynomial from the pair (p_0, p_1).

   Forming S-pol of p_0 and p_1:

   $$p_{411} = u_3x_2x_1 + (-u_3^2 - u_2^2 + 2u_2u_1)x_2 - u_2x_1^2$$

   S-pol added.

$$
\begin{aligned}
P_{H_a\_H_aH_a} &= 0 \text{ by the statement}\\
\left(\left(\left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot P_{H_aCH_a}\right) + \left(\frac{\overrightarrow{\_H_aC}}{\overrightarrow{BC}} \cdot P_{H_aBH_a}\right)\right) + \left(-1 \cdot \left(\left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot \frac{\overrightarrow{\_H_aC}}{\overrightarrow{BC}}\right) \cdot P_{BCB}\right)\right)\right) &= 0 \text{ by Lemma 32}\\
\left(\left(\left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot P_{H_aCH_a}\right) + \left(\left(-1 \cdot \frac{\overrightarrow{C\_H_a}}{\overrightarrow{BC}}\right) \cdot P_{H_aBH_a}\right)\right) + \left(-1 \cdot \left(\left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot \left(-1 \cdot \frac{\overrightarrow{C\_H_a}}{\overrightarrow{BC}}\right)\right) \cdot P_{BCB}\right)\right)\right) &= 0 \text{ by geometric simplifications}\\
\left(\left(\left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot P_{H_aCH_a}\right) + \left(-1 \cdot \left(\frac{\overrightarrow{C\_H_a}}{\overrightarrow{BC}} \cdot P_{H_aBH_a}\right)\right)\right) + \left(\frac{\overrightarrow{B\_H_a}}{\overrightarrow{BC}} \cdot \left(\frac{\overrightarrow{C\_H_a}}{\overrightarrow{BC}} \cdot P_{BCB}\right)\right)\right) &= 0 \text{ by algebraic simplifications}
\end{aligned}
$$

Success rates in proving Wernick's problems

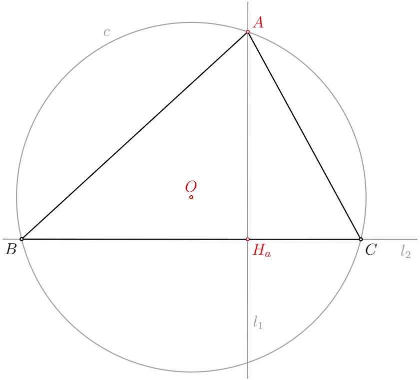
- We focused on problems from Wernick's corpus, solvable by ArgoTriCS
- Four different provers were used
  - provers integrated in GCLC (area method, Wu's method, and Gröbner bases method)
  - OpenGeoProver (Wu's method)
- OpenGeoProver was the most successful
- We worked also on portfolio approach, by choosing one particular solver for a specific correctness conjecture

Readability and understandability of proofs

- None of these correctness proofs don't look like classical, human-readable, synthetic proofs
- Readability is vital for educational purposes
- ArgoTriCS can be combined with CL provers to automatically obtain traditional, readable correctness proofs

Motivating example – readable proof

- Prove that A is the vertex of the constructed $\triangle ABC$, H_a is its altitude foot and O is its circumcenter



1. c_c contains vertices A, B, and C, so it must be the circumcircle of $\triangle ABC$
2. O is the center of c_c, so it must be the circumcenter of $\triangle ABC$
3. a contains the vertices B and C, so it must be the side BC of $\triangle ABC$
4. h_a contains A and is perpendicular to a, so it must be the altitude from vertex A
5. H_a belongs both to triangle side a and triangle altitude h_a, so it must be the altitude foot

Lessons following from previous example

- The previous correctness proof is readable
- It follows quite directly from the analysis: it just reverses the chain of deduction steps
- The proof relies on several uniqueness lemmas
- One could assume that proving correctness is always easy like this, however...
- ... in some cases the proof is quite different from the analysis

Automated generation of readable correctness proofs

- How can readable correctness proofs be automatically obtained?
- ArgoTriCS can automatically export the TPTP file containing:
  - the correctness conjecture
  - the set of axioms corresponding to the geometric definitions and lemmas
- It can be further passed to an automated theorem prover
- To be able to prove the correctness conjecture, apart from lemmas identified during development of ArgoTriCS, additional lemmas are needed
- Are there automated provers that can
  - prove these types of conjectures?
  - produce readable proofs?
  - possibly produce formal proofs?

Automated theorem provers based on coherent logic

- Coherent logic is a fragment of FOL convinient for producing readable proofs

$$A_1(\vec{x}) \wedge \ldots \wedge A_n(\vec{x}) \Rightarrow \exists \vec{y}_1\ \mathcal{B}_1(\vec{x}, \vec{y}_1) \vee \ldots \vee \exists \vec{y}_m\ \mathcal{B}_m(\vec{x}, \vec{y}_m)$$

- ArgoCLP is a coherent logic (CL) prover that can export both readable and formal proofs
- Larus is a CL prover that can export both readable and formal proofs (Janičić, Narboux)
- GCProver is a prover based on fragment of CL without disjunctions and existential quantifiers
- Correctness conjectures fall into this reduced fragment of CL, so each of these provers can be used

Example of automatically generated readable correctness proof

Axioms:

1. bc_unique : $\forall L\ (inc(pB, L) \land inc(pC, L) \Rightarrow L = bc\ )$
2. haA : $\forall H\ (perp(H, bc) \land inc(pA, H) \Rightarrow ha = H\ )$
3. pHa_def : $\forall H1\ (inc(H1, ha) \land inc(H1, bc) \Rightarrow H1 = pHa\ )$
4. cc_unique : $\forall C\ (inc\_c(pA, C) \land inc\_c(pB, C) \land inc\_c(pC, C) \Rightarrow C = cc\ )$
5. center_unique : $\forall C\ \forall C1\ \forall C2\ (center(C1, C) \land center(C2, C) \Rightarrow C1 = C2\ )$

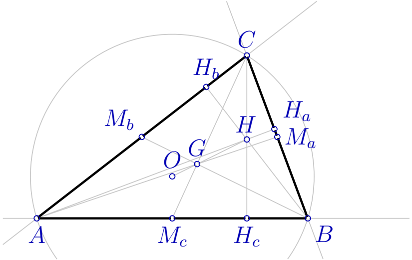
Theorem: th_A_Ha_O0 :
$inc(pA, ha1) \land inc(pHa1, ha1) \land perp(ha1, a1) \land inc(pHa1, a1) \land inc\_c(pA, cc1) \land center(pOc1, cc1) \land inc\_c(pB, cc1) \land inc(pB, a1) \land inc\_c(pC, cc1) \land inc(pC, a1) \Rightarrow pHa = pHa1$

Proof:

1. $pHa = pHa$ (by MP, using axiom eqnativeEqSub0; instantiation: $A \mapsto pHa$, $B \mapsto pHa$, $X \mapsto pHa$)
2. $a1 = bc$ (by MP, from $inc(pB, a1)$, $inc(pC, a1)$ using axiom bc_unique; instantiation: $L \mapsto a1$)
3. $perp(ha1, bc)$ (by MP, from $perp(ha1, a1)$, $a1 = bc$ using axiom perpEqSub1; instantiation: $A \mapsto ha1$, $B \mapsto a1$, $X \mapsto bc$)
4. $ha = ha1$ (by MP, from $perp(ha1, bc)$, $inc(pA, ha1)$ using axiom haA; instantiation: $H \mapsto ha1$)
5. $inc(pHa1, ha)$ (by MP, from $inc(pHa1, ha1)$, $ha = ha1$ using axiom incEqSub1; instantiation: $A \mapsto pHa1$, $B \mapsto ha1$, $X \mapsto ha$)
6. $inc(pHa1, bc)$ (by MP, from $inc(pHa1, a1)$, $a1 = bc$ using axiom incEqSub1; instantiation: $A \mapsto pHa1$, $B \mapsto a1$, $X \mapsto bc$)
7. $pHa1 = pHa$ (by MP, from $inc(pHa1, ha)$, $inc(pHa1, bc)$ using axiom pHa_def; instantiation: $H1 \mapsto pHa1$)
8. $pHa = pHa1$ (by MP, from $pHa1 = pHa$, $pHa = pHa$ using axiom eqnativeEqSub0; instantiation: $A \mapsto pHa$, $B \mapsto pHa1$, $X \mapsto pHa$)
9. Proved by assumption! (by QEDas)

Evaluation of effectiveness of Larus and GCProver

- The subset of Wernick's corpus is considered over:
  - vertices A, B, C
  - side midpoints M_a, M_b, M_c
  - feet of altitudes H_a, H_b, H_c
  - centroid G, circumcenter O and orthocenter H
- 35 non-isomorphic solvable problems



- Larus proved 20 problems, while GCProver proved all 35 correctness conjectures within the time-limit of 300s
- It often needed guidance to prove the correctness conjectures
- Unlike GCProver, Larus outputs also formal proofs in Coq and Isabelle/HOL

References

SPJ2011 S. Stojanović, V. Pavlović, P. Janičić,
A Coherent Logic Based Geometry Theorem Prover Capable of Producing Formal and Readable Proofs,
Automated Deduction in Geometry, ADG 2010,
volume 6877 of *Lecture Notes in Artificial Inteligence*, pp. 200–219. 2011.

M2015a V. Marinković,
Proof Simplification in the Framework of Coherent Logic,
Computing and Informatics (CAI), vol. 34, no.2, pp. 337–366, 2015.

NMKJ2019 M. Nikolić, V. Marinković, Z. Kovàcs, P. Janičić,
Portfolio Theorem Proving and Prover Runtime Prediction for Geometry,
Annals of Mathematics and Artificial Intelligence (AMAI),
vol. 85(2-4), pp. 119–146, 2019.

References

MSM2024 V. Marinković, T. Šukilović, F. Marić,
Towards Automated Readable Proofs of Ruler and Compass Constructions,
Electronic Proceedings in Theoretical Computer Science (EPTCS) 398, Proceedings of the 14th International Conference on Automated Deduction in Geometry, pp. 11–20, 2024.

N2024 V. Novaković,
Design and implementation of the prover of conjectures of correctness of geometric constructions (in Serbian language),
Master thesis, Faculty of Mathematics, University of Belgrade, Serbia, 2024.

MSNM2025 V. Marinković, T. Šukilović, V. Novaković, F. Marić:
Readable automated proofs of ruler and compass constructions,
Annals of Mathematics and Artificial Intelligence (AMAI), 2025.

Table of Contents

Status of proofs

- Correctness of background geometric knowledge used by ArgoTriCS has, until now, been taken for granted
- Obtained correctness proofs currently rely on high-level lemmas

```
fof(ratio23_AGAMa, axiom,(ratio23(pA,pG,pA,pMa))).
```

- To be sure in validity of those proofs, correctness of used lemmas should be proved

Formalization of lemmas

- We are working on formalization of common geometric lemmas used as axioms within GCProver (Pierre Boutry)
- Lemmas are being proved from the Tarski's axioms of Euclidean geometry, using GeoCoq
- Up to now, 55 out of 144 lemmas are proved
- Most of the remaining 89 lemmas are about:
  - ratio of lengths of directed segments
  - equality of non-primitive objects, such as lines and circles
  - asserting the existence of geometrical objects
- Work in progress

References

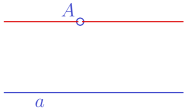
BMM2025 P. Boutry, V. Marinković:
Report from meeting “Rule based automated provers for geometry” organized within EuroProofNet COST action CA20111, 2024.

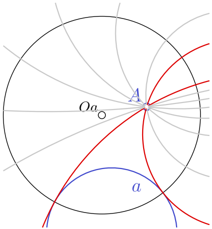
BMM2025 P. Boutry, V. Marinković, F. Marić:
Report on STSM “Formal verification of rule-based geometry” organized within EuroProofNet COST action CA20111, 2025.

Table of Contents

Different geometries

- Until now we considered the most common type of geometry – Euclidean geometry
- Absolute geometry is based on four axioms groups: incidence, order, congruence, and continuity
- By adding the appropriate axiom of parallelism, we get:
  - Euclidean geometry: a unique line parallel to a line a through a point A not on the line
  - Hyperbolic geometry: infinitely many parallels to a line a through a point A not on the line

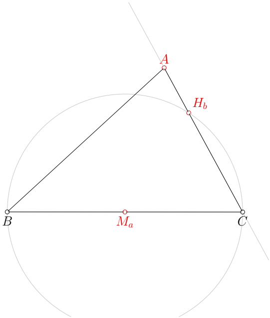


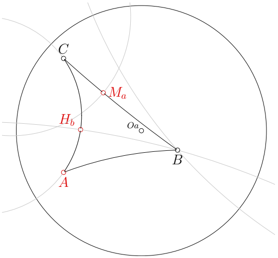


- Challenge: automatically find constructions in absolute and hyperbolic geometry.

Automating constructions in hyperbolic geometry

- ArgoTriCS was initially focused solely on Euclidean geometry
- Many ruler and compass constructions are valid only in Euclidean geometry

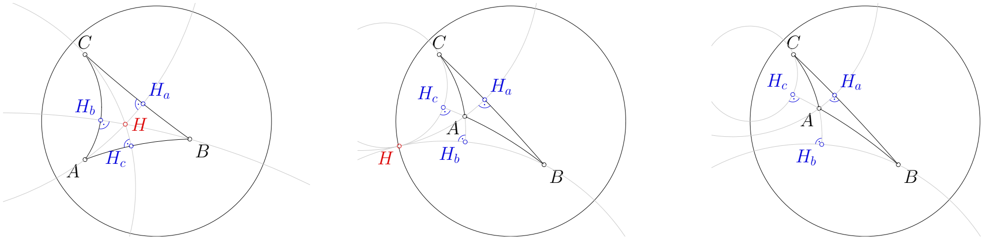




- ArgoTriCS can be adjusted to hyperbolic geometry by solely adapting geometric knowledge
- The first study on automated triangle constructions in hyperbolic geometry

Differences in geometric knowledge between Euclidean and hyperbolic case

- The centroid of the triangle does not divide the median in 2:1 ratio
- The inscribed angle subtended by a diameter need not be right
- The circumcenter and the orthocenter of a triangle need not exist



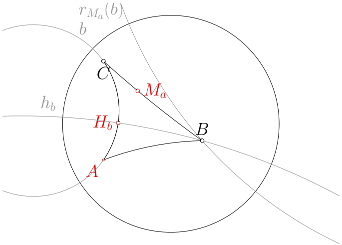
- Reflections play the most important role in hyperbolic solving

Example construction in hyperbolic geometry

Problem: Construct the triangle ABC given the vertex A, side midpoint M_a, and altitude foot H_b.

Construction:

1. Construct the line b through the points A and H_b;
2. Construct the line h_b perpendicular to the line b through H_b;
3. Construct the line $s_{M_a}(b)$ that is image of the line b under the reflection wrt. point M_a;
4. Construct the intersection point B of the lines $s_{M_a}(b)$ and h_b;
5. Construct the point C symmetric to B wrt. point M_a.



Evaluation of Construction Generation in Hyperbolic Geometry

- We considered the Wernick's corpus of construction problems
- From 139 significantly different problems, ArgoTriCS solved 31 solvable problems, 1 redundant and 11 locus dependent
- Automatically generated compendium of solutions to problems in hyperbolic geometry: http://poincare.matf.bg.ac.rs/~vesnap/animations_hyp/compendium_wernick_hyperbolic.html

References

MSM2021 V. Marinković, T. Šukilović, F. Marić:
On automating triangle constructions in absolute and hyperbolic geometry,
Electronic Proceedings in Theoretical Computer Science (EPTCS) 352,
Proceedings of the 13th International Conference on Automated Deduction in Geometry,
pp. 14–26, 2021.

MSM2023 V. Marinković, T. Šukilović, F. Marić:
Automated triangle constructions in hyperbolic geometry,
Annals of Mathematics and Artificial Intelligence (AMAI), vol. 91, pp. 821–849, 2023.

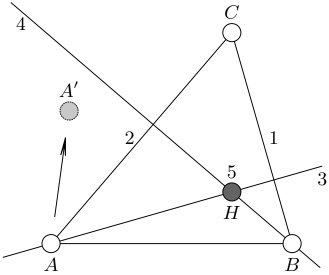
Table of Contents

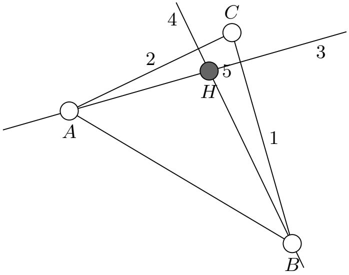
Dynamic geometry tools

- Dynamic geometry tools appeared at late 20th century
- Provoked big changes in educational practices:
  - Enables visualisation
  - Allow real-time experimentations
- Through years they were integrated with automated theorem proving tools used for validation of properties, proof verification, etc.
- They became an important component of mathematical education

Typical interactivity in dynamic geometry tools

- Usual scenario:
  - user chooses several free points
  - using them user constructs other geometric objects
  - user can move free point and explore how other (constructed) objects change by recomputing coordinates of constructed points



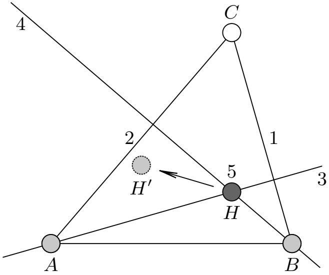


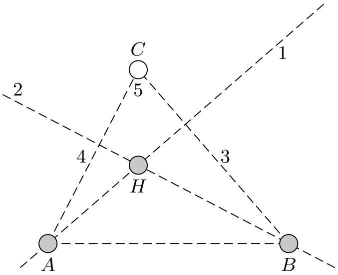
Moving constructed points

- Rather new scenario: moving constructed point
  - necessary to specify which points stay fixed
  - user can move only one of the significant points in triangle
  - reconstruction of location of all other objects in the triangle is performed by reconstruction of the coordinates of triangle vertices
- Reconstruction of triangle vertices given positions of significant points in triangle boils down to solving triangle construction problem, performed by ArgoTriCS
- This scenario is implemented in tool Touch&Drag for touch screens

Example of moving constructed points

- Vertices A, B and C are chosen freely, orthocenter H of $\triangle ABC$ is constructed
- Orthocenter H is moved, while vertices A and B stay fixed
- How to determine a new position of vertex C?





References

S2018 M. Selaković,
Geometric constructions on touch-sensitive devices (in Serbian language),
Master thesis, Faculty of Mathematics, University of Belgrade, Serbia, 2018.

SMJ2020 M. Selaković, V. Marinković, P. Janičić,
New Dynamics in Dynamic Geometry: Dragging Constructed Points,
Journal of Symbolic Computation, vol 97, pp. 3–15, 2020.

Table of Contents

Next step guidance

- Help students once they get stuck and do not know how to proceed with the construction
- Enables interacivity and keeps student active participant of the learning process
- Different types of help:
  - suggest the next construction step
  - suggest an object (point, line, circle) that should be constructed
  - suggest a lemma that the construction relies on

Target construction

- There can be many possible constructions for the same problem
- Not all solutions are intuitive and easily comprehendable
- Example: A, B, H
  - A, B, H - solution 1
  - A, B, H - solution 2
- Next step guidance assumes that we select one target construction that is closest to the partial construction given by the student
- There are different approaches to choose it:
  - ArgoTriCS can find several possible constructions (in advance), compare them to the partial construction, and select the most adequate one
  - ArgoTriCS can be restarted given all the objects constructed by the partial construction

Suggestions

- Once the target construction is fixed, what should be suggested?
  - the next construction step
    - A, B, H - solution 1
      Suggestion: "construct a perpendicular to a known line through a known point"
  - an important auxiliary object (point, line, circle)
    - A, B, H - solution 2
      Suggestion: "construct circle over segment AB"
  - an important lemma used in construction correctness proof
    - A, B, H - solution 1
      "altitudes are orthogonal to triangle sides ($h_a \perp a$, $h_b \perp b$)".

References

MM2021 V. Marinković, F. Marić,
Towards Next Step Guidance in Triangle Construction Problems,
Workshop on Theorem proving components for Educational software (ThEdu) 2021,
Pittsburgh, United States (virtual)

Table of Contents

Conclusions

- At the intersection of all topics discussed is ArgoTriCS – a system for automating constructions
- Its focus in primarily on educational aspect, but keeping a level of rigor
- It is adaptable
  - can be used for both Euclidean and hyperbolic geometry
  - usually starts from three located points, but also can start from different number of different objects or non-locational data
  - can be integrated with algebraic provers, but also FOL and CL provers, as well as interactive theorem provers
  - can cooperate with dynamic geometry software
- Plans for the future: finishing NSG and lemma formalization, adding full suport for LD problems, experimenting with origami

Thanks

- Thank you for your attention!