ADG-Lib Initiative

ADG-Lib Initiative Team[1]

[1]Pedro Quaresma, University of Coimbra, Portugal, pedro@mat.uc.pt — Predrag Janičić, University of Belgrade, Serbia, janicic@matf.bg.ac.rs — Julien Narboux, Université Paris Cité, IRIF, France, narboux@irif.fr — Zoltán Kovács, Private University of Education, Diocese Linz, Austria , zoltan.kovacs@ph-linz.at — Anna Petiurenko, Pedagogical University of Kraków, Poland, anna.petiurenko@uken.krakow.pl — Filip Marić, University of Belgrade, Serbia filip@matf.bg.ac.rs — Nuno Baeta, University of Coimbra, Portugal, nmsbaeta@gmail.com

ADG 2025, 1-2 August, Stuttgart, Germany

The Need for a Common Ground

Over the last 70 years, many resources related to automated reasoning in geometry has been developed: axiom systems, sets of conjectures, and tools.

But almost each of them has its own language in which statements are expressed, which prevents the sharing of resources.

Because of that, the need for a common library of axiomatic systems and theorems for geometric automated theorem provers have been felt for many years now.

Under the umbrella of the *EuroProofNet* COST action, we meet at Nice, France, on 28–30 May 2024 (WG2-GEO24) and in Kraków, Poland, on June 3–5, 2025 (WG2-GEO25), to push forward this unifying agenda.

Motivations

Why do we need something different from general-purpose automatic theorem provers ?

- GATPs are designed specifically for deciding *certain* statements in *certain* geometries.
  - Which geometry? (neutral/Euclidean/Hyperbolic, 2D/3D/nD, ordered or not . . .)
  - Which set of statements? (constructive or not, . . .)
- Geometric conjectures may not be correct (non-degeneracy conditions are often implicit)

Why do we advocate for a standard signature?

- Semantics of common geometric predicates may differ from one prover to another.
- A standard signature and library of conjecture will help using machine learning for automatic formalization, or automatic theorem proving.

Example

Algebraic provers (Wu's method, or Gröbner basis), and the Chou, Gao and Zhang's Area Method, assume that $AA \parallel CD$ always holds, whereas some other provers may assume that only non-degenerate lines are parallel.

For the area method, the situation is even more subtle since the meaning of parallel in assumptions and goal differ.

The ADG-Lib initiative

Encouraged by the success of the SMT-Lib initiative, which significantly advanced the field of SMT.

We set similar goals for the ADG-Lib initiative to foster advances in the field of Automated Deduction in Geometry.

- Provide a rigorous description of the axiom systems used by the different geometry, automatic and interactive theorem provers
- Propose a common input and output language for geometry theorem provers
- Establish and make available to the research community a large library of geometric problems.

ADG-Lib Signature

At the *WG2-GEO25* meeting, we discussed the preliminary ADG-Lib signature and resolved a number of open issues:

- underlying theories — the first one is solid Euclidean geometry, with one sort: points;
- naming conventions — CamelCase notation; verbose — long identifiers (non-ambiguous).
- sets of predicate symbols, function symbols, and sets of non-deterministic function symbols (such as "onRay");
- NDG (non-degenerate cases) and DG (allowing degenerate cases) versions of predicate symbols;
- other issues: order of arguments, names of arguments, redundancies, etc.

The plan is to port the GeoCoq library of formal geometry proofs to ADG-Lib, when done, this will be version 1.0 of ADG-Lib.

ADG-Lib Signature

Points-only; predicate symbols

between(A,B,C)	$A{-}B{-}C$	points A, B and C are collinear and $\overrightarrow{AB} \subsetneq \overrightarrow{AC}$	
collinear(A,B,C)	Col $A\,B\,C$	points A, B and C are collinear	$A{-}B{-}C \vee B{-}A{-}C \vee A{-}C{-}B$
congruent(A,B,C,D)	$AB \equiv CD$	the segments AB and CD are congruent (intuitively in the sense that they have same length, but length measure is not assumed to exist)	
midpoint(M,A,B)	$A{-}M{-}B$	M is the midpoint of segment AB	$A{-}M{-}B \wedge AM \equiv BM$
perpendicular(A,B,C,D)	$AB \perp CD$	line AB is perpendicular to line CD ($A \neq B$ and $C \neq D$)	$\exists P, AB \underset{P}{\perp} CD$
parallel(A,B,C,D)	$AB \parallel CD$	line AB is parallel to line CD	$AB \parallel_s CD \vee (A \neq B \wedge C \neq D \wedge$ Col $A\,C\,D \wedge$ Col $B\,C\,D)$

Choice of a language

- Most geometrical statements are expressible in first-order logic (often in coherent logic).
- We chose to reuse the TPTP/FOL standard for compatibility with general-purpose ATP.

ADG-Lib Example

Example of a file with a conjecture:

```
include('../Axioms/gelernter.ax').
fof(thmB, conjecture, ![A,B,C,D]: (
    (
     ~collinear(A,B,D) & ~collinear(C,D,B) & parallel(B,C,A,D)
     & congruent(B,C,A,D) & oppositeSides(D,B,A,C)
    )
    =>
    congruent(A,B,C,D)
    )).
```

- TPTP/FOF based.
- Inclusion of a set of axioms (`gelernter.ax`).
- All points are universally quantified.
- Predicate symbols from ADG-Lib

ADG-Lib Signature

- predicate symbols
- predicates on coordinates
- deterministic function symbols
- non-deterministic function symbols
- Area method; specific predicates and function symbols
- Space geometry; predicate symbols

ADG-Lib Converters

At the *WG2-GEO25* meeting, following earlier, available *C* and *Prolog* parsers for conjectures stored in *GCLC*, *JGEx*, and *GeoGebra* formats, we have developed a completely new, more robust general converter toolkit, the *ADG-Lib-toolkit*. The following components were developed:

- parsers for *GCLC* and *JGEx* formats;
- data-structures for internal representation following *ADG-Lib*;
- output for *GCLC*, *GeoGebra*, *TPTP* formats.

ADG-Lib Converters

The plan is to implement also:

- parsers for *GeoGebra* and *TPTP* formats, and output for *JGEx*;
- a LaTeX package defining some standard notation for common predicates;
- other signature based on additional types (lines, planes, angles, . . .)
- a tool for conversion between point only and point/line/plane signatures;
- a tool for conversion from declarative conjecture representation into a procedural counterpart;
- a tool to avoid duplications in future repositories of problems ([7]).

ADG-Lib Axioms Sets and Conjectures

During the *WG2-GEO25* meeting, using the *ADG-Lib-toolkit*, we started to build a set of geometry conjectures by converting available *GCLC* conjectures.

However, we also started to encode (in the *TPTP* version of *ADG-Lib*) available axiom systems and accompanying conjectures, for now — by Gelertner [6] and by Flenner [5].

To be also included, the set of problems already in the Geometry section of *TPTP*, the problems by Chou et al. ([1, 2, 3, 4]), and the sets of problems in *TGTP*, *GCLC*, *GeoGebra*, *JGEx*.

ADG Foundation

The *ADG-Lib* initiative should be framed under the overall goal of establishing a community of researchers/practitioners in Automated Deduction in Geometry.

ADG Foundation

The non-profit association aims to promote science and technology in the field of computer supported reasoning in geometry by:

- Representing the interests of those involved in research and teaching,
- promoting knowledge and application of algorithms and software for automated geometric reasoning,
- developing and expanding the subject area.

A key objective of the association is to organise the ADG (Automated Deduction in Geometry) conference.

Web & GitHub Pages & How to Contribute

Web-page: adg-foundation.info

GitHub page: github.com/ADG-Foundation

Two main, open source, projects:

- ADG-Lib (documentation, *ADG-Lib-toolkit*, repository of problems);
- ADG-Provers (stable versions of the supported provers that are under the umbrella of ADG-Foundation).

How to contribute:

- Join the ADG Foundation.
- Join the ADG-Lib project.

Web & GitHub Pages & How to Contribute

Web-page: adg-foundation.info

GitHub page: github.com/ADG-Foundation

Two main, open source, projects:

- ADG-Lib (documentation, *ADG-Lib-toolkit*, repository of problems);
- ADG-Provers (stable versions of the supported provers that are under the umbrella of ADG-Foundation).

How to contribute:

- Join the ADG Foundation.
- Join the ADG-Lib project.

Thank You

Bibliography I

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang.

A collection of 110 geometry theorems and their machine produced proofs using full-angles.

Technical Report TR-94-4, Department of Computer Science, The Wichita State University, November 1994.

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(3):325–347, dec 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(3):349–370, dec 1996.

Bibliography II

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(3):219–246, 2000.

Patricia O'Neil Flenner.

A subset of gelernter's geometry theorem prover.

Master Thesis, California State University, 1983.

Herbert L. Gelernter.

Realization of a geometry theorem proving machine.

In *Information Processing, Proceedings of the 1st International Conference on Information Processing, UNESCO, Paris 15-20 June 1959*, pages 273–281. UNESCO (Paris), 1959.

Bibliography III

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.