Towards Ranking Geometric Automated Theorem Provers

Nuno Baeta
CISUC
University of Coimbra, Portugal
nmsbaeta@gmail.com

Pedro Quaresma
CISUC / Department of Mathematics
University of Coimbra, Portugal
pedro@mat.uc.pt

The field of geometric automated theorem provers has a long and rich history, from the early AI approaches of the 1960s, synthetic provers, to today algebraic and synthetic provers.

The geometry automated deduction area differs from other areas by the strong connection between the axiomatic theories and its standard models. In many cases the geometric constructions are used to establish the theorems' statements, geometric constructions are, in some provers, used to conduct the proof, used as counter-examples to close some branches of the automatic proof. Synthetic geometry proofs are done using geometric properties, proofs that can have a visual counterpart in the supporting geometric construction.

With the growing use of geometry automatic deduction tools as applications in other areas, e.g. in education, the need to evaluate them, using different criteria, is felt. Establishing a ranking among geometric automated theorem provers will be useful for the improvement of the current methods/implementations. Improvements could concern wider scope, better efficiency, proof readability and proof reliability.

To achieve the goal of being able to compare geometric automated theorem provers a common test bench is needed: a common language to describe the geometric problems; a comprehensive repository of geometric problems and a set of quality measures.

1 Introduction

The first geometric automated theorem provers proposed, that came as early as in 1959 [13], adapt general-purpose reasoning approaches developed in the field of artificial intelligence, automating the traditional geometric proving processes. In order to avoid combinatorial explosion while applying postulates, many suitable heuristics, e.g. adding auxiliary elements to the geometric configuration, have been developed. Although being able to produce readable proofs, the proposed methods were very narrow-scoped and not efficient [34].

Recent results using this approach, like the *ArgoCLP* theorem prover, based on coherent logic [30], or the deductive database approach [11] having been proposed with different rates of success in different classes of geometric problems.

The algebraic methods, such as the characteristic set method [5, 34], the polynomial elimination method [32], the Gröbner basis method [18], and the Clifford algebra approach [20], reduce the complexity of logical inferences by computing relations between coordinates of geometric entities. What is gained in efficiency and wider scope is lost in the connection of the algebraic proof and the geometric reasoning. These methods are broad-scope, efficient, but if, eventually, a proof record is produced, it will be a very complex algebraic proof [7, 34]. This is still an active area of research [2, 14, 15, 19, 37].

In order to combine the readability of synthetic methods and efficiency of algebraic methods, some approaches, such as the area method [9, 14], the full angle method [10] represent geometric knowledge in a form of expressions with respect to geometric invariants. These methods are broad-scoped (less

Pedro Quaresma, Walther Neuper (Eds.): Theorem proving
components for Educational software (ThEdu'18)
EPTCS 290, 2019, pp. 30–37, doi:10.4204/EPTCS.290.3

than the algebraic), efficient (less than the algebraic) and capable of producing readable proofs. An implementation as a *Coq*[1] contribution is such that it can have theirs proofs verified [3, 14].

When considering ranking the Geometric Automated Theorem Provers (GATP) we have to consider some, somehow opposing, goals: scope, the geometries being considered and what kind of problems are provable; efficiency, the time needed to complete the proof; readability of the proof produced (if any); and, with the help of proof assistants, the reliability of generated proofs [17, 24].

To be able to compare the different methods and implementations, a test bench must be defined, a common language to state the geometric theorems, a comprehensive repository of geometric problems and a set of measures of quality capable of assessing the GATPs in different classes, such as: scope; efficiency; readability; reliability of generated proofs.

Overview of the paper. The paper is organised as follows: first, in Section 2, the test bench is discussed. In Section 3, the different measures of quality are discussed. Final conclusions are drawn and future work is foreseen in Section 4.

2 Test Bench

In order to implement a test bench, the Thousands of Geometric problems for geometric Theorem Provers[2] (*TGTP*) platform presents itself as a solid foundation to fulfil such purpose. *TGTP* aims to provide the automated reasoning in geometry community with a comprehensive and easily accessible, library of GATP test problems [23], it already provides a centralised common repository of geometric problems[3], an unambiguous reference mechanism, textual and geometric search mechanism and the problems are kept in the I2GATP common format [25]. Moreover, *TGTP* provides, as part of its infrastructure, implementations of several methods, namely: *GCLC*[4] implementations of Wu's method, Gröbner basis method and the area method; and a *Coq* implementation of the area method[5] [21]. Statistical and performance information is also supplied for all implementations, as well as a proof status for each geometric problem.

With a working test bench in operation, an interesting goal to pursue would be to operationalize a competition between GATPs, similar to *CASC* [31]. Its ultimate goal, like in *CASC*, would be to encourage researchers to improve existing GATPs and implement new ones.

For example, in table 1 the CPU times[6] needed to complete the proofs for some *TGTP* problems are presented. The GATPs used were: *Coq*;[7] *GCLC*; *GeoGebra*; *OpenGeoProver* (OGP) with the methods: *area method* (AM); *Wu's method* (WM); *Gröbner Basis' method* (GBM); *BotanaGiac* (BG) [1]. The *OGP* and *GCLC* implementations of the Wu's method stand-off as the only implementations that were able to prove all the conjectures, with *GCLC* having a marginal advantages over *OGP*.

But, apart CPU times, when ranking GATPs other issued must be considered: the scope, the proof readability and proof reliability.

The *GCLC* implementation of the area method is the only GATP providing a readable synthetic geometric proof.

[1] `https://coq.inria.fr/`

[2] `http://hilbert.mat.uc.pt/TGTP/`

[3] As of 2018–04–21 there were 236 problems.

[4] `http://poincare.matf.bg.ac.rs/~janicic/gclc/`

[5] `https://github.com/coq-contribs/area-method`

[6] Linux 4.9.0-2-amd64 #1 SMP Debian 4.9.18-1 (2017-03-30), Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz. The used "time-out" threshold was 60s, but it can be configured for any given value.

[7] For some of the examples it was not possible to have the transcription of the problem in *Coq*

	Coq	**GCLC**			**GeoGebra**	**OGP**
	AM	AM	WM	GBM	BG	WM
GEO0080	0.74	0.012	0.016	time-out	0.615	0.028
GEO0226	—	0.748	1.62	time-out	time-out	0.075
GEO0228	—	0.008	0.012	0.036	0.207	0.015
GEO0237	time-out	time-out	0.048	time-out	0.264	0.105
GEO0238	time-out	0.032	0.024	0.092	0.18	0.102
GEO0240	—	time-out	0.056	time-out	1.314	0.030

Table 1: Comparing GATPs

The *GCLC* implementation of the Wu's method is the one with the wider scope (for this small set of examples), being able to prove all the problems.

The *Coq* implementation of the area method is the only GATP formally verified [3, 14]. The *Coq* proof assistant relies on a small kernel, extensively verified, against whom all the other layers are formally checked. The *Coq* implementation of the area method is thus formally verified. For all the other GATPs the implementations trustworthiness relies on the GATPs code as a whole, without any formal verification of its correctness. This issue is more important when the algebraic implementations are to be considered, in those cases its black box nature, with only a yes/no answer, prevent the user to manually check the proof produced. Any new implementation should be extensively checked before it can be considered finished.

The intended use is also important. For example, in education two uses of a GATP are foreseen, the validation of a given conjecture and/or construction [1, 16] and the proof itself as object of study.

For the first case (validation), the proof is not needed, only a formal validation is needed, speed is the most important measure of quality. Rowe [28] and, later, Stahl [29] studied this issue, defining the "Post-Teacher Question Wait-Time", i.e. the time a high-school class will wait in silence after a teacher place a question. This is a (surprisingly!?) short span of time, a good response time is less then 1.5s, a fair response time will be less then 3s, everything above that would be unsuitable for a high-school classroom. Looking in table 1, we can see that the implementations of the algebraic methods (with the exception of *GCLC* implementation of the Gröbner basis) gave good results for all but two cases. One result in the border between good and fair and a "time-out".

For the second case a readable proof is needed, i.e. a synthetic or semi-synthetic geometric proof readable, by teachers (at least) and students. For the first case the algebraic methods seems to be the best one, for the second case they are completely useless, the proofs produced are algebraic, unrelated with the geometric construction in consideration. The synthetic or even semi-synthetic should be considered instead.

As far as the authors of this paper know, there are two proposals to measure the readability of a proof: Chou et al. [8] proposed a way to measure how difficult a formal proof is (using the area method) based on the time and the number of steps needed to perform it; de Bruijn proposed the de Bruijn factor [4, 33],[8] the quotient of the *size* of corresponding informal proof and the *size* of the formal proof. Apart this quantitative measures some qualitative considerations and also considerations about the audience in consideration led to a schema, proposed in [26], to classify the proofs produced by GATPs:

1. no readable proof;

[8] `https://www.cs.ru.nl/~freek/factor/`

2. non-synthetic proof (i.e. a proof without a correspondent geometric description, e.g. algebraic methods);
3. semi-synthetic proof with a corresponding prover's language rendering;
4. (semi-)synthetic proof with a corresponding natural language rendering;
5. (semi-)synthetic proof with a corresponding natural language and visual renderings [35, 36].

The synthetic proofs with a corresponding natural language and visual renderings is the most difficult to attain, but also the more desirable, e.g. in an educational setting.

3 Ranking GATP

To be able to compare the different methods and implementations, alongside a standard test bench (see Section 2), we must define a set of measures of quality capable of assessing the GATPs in different classes: scope; efficiency; readability; reliability of generated proofs.

Scope To measure the scope of a GATP, one should consider:

- which geometries are allowed by the GATP—despite the fact that most GATPs deal only with Euclidean geometry, some exist that prove geometric problems in non-Euclidean geometries [6, 38];
- what kind of problems are provable—as an example, the area method uses *geometric invariants* as basic quantities to prove theorems, each of which is used to deal with different geometric relations, hence allowing certain theorems to be easily proved [8].

Considering the existing methods, algebraic ones have the broadest scope. Not only have they been used to prove theorems in Euclidean and non-Euclidean geometry, but, for every geometry, the range of difficulty of the problems proved is very wide [6]. The earlier synthetic approaches were narrow scoped, e.g. the *GEOM* [12] only dealt with a limited set of geometric elements and relations [8]. In between lie semi-synthetic methods and coherent logic based methods. Semi-synthetic methods, starting with the area method which is complete for *constructive geometry* [9, 10, 14], with its *geometric invariant*, the *signed area*, allow many problems with relations like incidence and parallelism to be proved. Adding another *geometric invariant*, the *Pythagoras difference*, allows the problems with relations like perpendicularity and congruence of line segments to be easily proved. Adding other *geometric invariants* such as the *full-angle* (which gives its name to the full-angle method), the *volume* and the *vector*, allows the demonstration of an ever increasing range of theorems [8].

Synthetic and semi-synthetic methods scope may be influenced by the use of deductive databases [11, 12]. Indeed, as stated in [11], unexpected results may be obtained, some of which are possibly new.

Efficiency The purpose of a GATP is, in addition to prove geometric conjectures, that these are obtained efficiently. By and large, although other resources may be involved, efficiency is related to time and memory space: we look for algorithms/implementations capable of fulfilling a proof in a reasonable amount of time and space.

Time is indeed the *natural* way to measure efficiency since it is used extensively, if not exclusively, throughout the literature [6, 7, 9, 10, 11, 14]. and, for obvious reasons, in a competition such as

CADE ATP System Competition (*CASC*). Moreover, such measure is of paramount importance when considering an educational environment.

Note however that when authors state something about GATP times, they do so in their settings, i.e. computer and operating system used, somehow restricting the usefulness of these results. The existence of a free and open platform where different GATPs can be tested on equal terms proves to be of utmost importance. In table 1 the examples were tested under the same system being comparable, an ongoing work will allow using all the conjectures in *TGTP* to test all those GATPs under the same system.

Regarding space, the authors are unaware of any study, presumably because these physical constraints are nowadays less important. Besides, from the users point of view time is the most important factor.

Readability Until recently, proofs in mathematics were solely made and verified by humans. With the advent of computers and automated reasoning that is no longer the case. As in some settings readability of a proof (by a human) is of utmost importance, such characteristic is considered crucial. Indeed, in an educational setting the proof is an object of learning by itself and the ability of a GATP to produce a synthetic proof, with the usual geometric inference rules, is of fundamental importance to its usability.

TGTP both the quantitative and qualitative measures should be introduce in such a way that a user can filter the information in the way that best fit his/her needs.

Reliability By reliability is meant the confidence that we have in the proofs made by a given prover. Is the prover correctly implemented?

Using proof assistants like *Coq*, or *Isabelle*,[9] the implementation of a given prover can be formally established, e.g. in [21] a implementation of the area method within *Coq* is described, where all the properties of the geometric quantities required by the area method are verified, demonstrating the correctness of the system, reducing concerns of reliability, to the trustworthiness of respective proof assistants [14, 21].

For the other implementations, the trustworthiness of GATP relies on the implementation code as a whole and in the extensive testing of the new implementations, e.g. using a workbench like *TGTP*.

4 Conclusion and Future Work

When regarding the area of geometric automated deduction the evaluation of automated provers is no longer a mono-objective problem, time considerations are important but scope and readability of the proofs are also very important.

Maybe the question “what is the best GATP of them all”, can not be answered, but at least we should have some partials answers when looking for a GATP that fit a particular goal.

The intended audience is also a point to be considered, e.g the narrow scope provers with specific heuristics, closing the circle with the early AI approaches of the 1960s, may be not relevant to an expert in the field of automated theorem proving, but the definition of sub-languages for educational use, could led to narrow scope GATPs with good qualities when readability of the proofs is to be considered, in specific educational settings.

[9] `https://isabelle.in.tum.de/`

We need a common workbench: a common language to describe the geometric problems; a large set of problems and a set of measures. Building on *TGTP* test bench [23], I2GATP common language [25] and *CASC*, CADE ATP System Competition [31], the first two need to be improved and better integrated, and adapting the ideas of the last, a common test bench should be built, with the goal of helping GATPs developers to improve their systems, to better fit the needs of the different communities of users.

References

[1] Francisco Botana, Markus Hohenwarter, Predrag Janičić, Zoltán Kovács, Ivan Petrović, Tomás Recio & Simon Weitzhofer (2015): *Automated Theorem Proving in GeoGebra: Current Achievements*. Journal of Automated Reasoning 55(1), pp. 39–59, doi:10.1007/s10817-015-9326-4.

[2] Francisco Botana & Pedro Quaresma, editors (2015): *Automated Deduction in Geometry, 10th International Workshop, ADG 2014*. Lecture Notes in Artificial Intelligence 9201, Springer. doi:10.1007/978-3-319-21362-0.

[3] Pierre Boutry, Julien Narboux, Pascal Schreck & Gabriel Braun (2014): *Using small scale automation to improve both accessibility and readability of formal proofs in geometry*. In Francisco Botana & Pedro Quaresma, editors: Preliminary Proceedings of the 10th International Workshop on Automated Deduction in Geometry, ADG 2014, Coimbra, Portugal, 9–11 July, 2014, CISUC Tech Reports TR 2014/01, pp. 31–49. Available at `https://www.cisuc.uc.pt/ckfinder/userfiles/files/TR2014-01.pdf`.

[4] Nicolaas Govert de Bruijn (1994): *A Survey of the Project Automath*, chapter A Survey of the Project Automath, pp. 141–161. Studies in Logic and the Foundations of Mathematics 133, North Holland, doi:10.1016/S0049-237X(08)70203-9.

[5] S.C. Chou (1985): *Proving and discovering geometry theorems using Wu's method*. Ph.D. thesis, The University of Texas, Austin.

[6] Shang-Ching Chou (1988): *Mechanical Geometry Theorem Proving*. Mathematics and Its Applications 41, D. Reidel Publishing Company.

[7] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (1994): *A Collection of 110 Geometry Theorems and Their Machine Produced Proofs Using Full-Angles*. Technical Report TR-94-4, Department of Computer Science, Wichita State University. Available at `https://www.researchgate.net/publication/239564904`.

[8] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (1994): *Machine Proofs in Geometry: Automated Production of Readable Proofs for Geometry Problems*. Applied Mathematics 6, World Scientific, doi:10.1142/9789812798152. Available at `https://www.researchgate.net/publication/240102887`.

[9] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (1996): *Automated Generation of Readable Proofs with Geometric Invariants: I. Multiple and Shortest Proof Generation*. Journal of Automated Reasoning 17(3), pp. 325–347, doi:10.1007/bf00283133.

[10] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (1996): *Automated Generation of Readable Proofs with Geometric Invariants: II. Theorem Proving With Full-Angles*. Journal of Automated Reasoning 17(3), pp. 349–370, doi:10.1007/BF00283134.

[11] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (2000): *A Deductive Database Approach to Automated Geometry Theorem Proving and Discovering*. Journal of Automated Reasoning 25(3), pp. 219–246, doi:10.1023/A:1006171315513.

[12] Helder Coelho & Luis Moniz Pereira (1986): *Automated Reasoning in Geometry Theorem Proving with Prolog*. Journal of Automated Reasoning 2(4), pp. 329–390, doi:10.1007/BF00248249.

[13] H. Gelernter (1995): *Realization of a geometry-theorem proving machine*. In: Computers & thought, MIT Press, Cambridge, MA, USA, pp. 134–152, ISBN: 0-262-56092-5.

[14] Predrag Janičić, Julien Narboux & Pedro Quaresma (2012): *The Area Method: A Recapitulation*. *Journal of Automated Reasoning* 48(4), pp. 489–532, doi:10.1007/s10817-010-9209-7.

[15] Predrag Janičić & Pedro Quaresma (2006): *System Description: GCLCprover + GeoThms*. In Ulrich Furbach & Natarajan Shankar, editors: *Automated Reasoning: Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17–20, 2006, Proceedings*, *Lecture Notes in Artificial Intelligence* 4130, Springer, pp. 145–150, doi:10.1007/11814771_13.

[16] Predrag Janičić & Pedro Quaresma (2007): *Automatic Verification of Regular Constructions in Dynamic Geometry Systems*. In Francisco Botana & Tomás Recio, editors: *Automated Deduction in Geometry: 6th International Workshop, ADG 2006, Pontevedra, Spain, August 31–September 2, 2006, Revised Papers*, *Lecture Notes in Artificial Intelligence* 4869, Springer, pp. 39–51, doi:10.1007/978-3-540-77356-6_3.

[17] Jianguo Jiang & Jingzhong Zhang (2012): *A review and prospect of readable machine proofs for geometry theorems*. *Journal of Systems Science and Complexity* 25(4), pp. 802–820, doi:10.1007/s11424-012-2048-3.

[18] Deepak Kapur (1986): *Using Gröbner bases to reason about geometry problems*. *Journal of Symbolic Computation* 2(4), pp. 399–408, doi:10.1016/S0747-7171(86)80007-4.

[19] Zoltán Kovács (2015): *The Relation Tool in GeoGebra 5*, pp. 53–71. *Lecture Notes in Artificial Intelligence* 9201, Springer International Publishing, doi:10.1007/978-3-319-21362-0_4.

[20] H. Li (2000): *Clifford algebra approaches to mechanical geometry theorem proving*. In X.-S. Gao & D. Wang, editors: *Mathematics Mechanization and Applications*, Academic Press, San Diego, CA, pp. 205–299, doi:10.1016/B978-012734760-8/50009-0.

[21] Julien Narboux (2009): *Formalization of the Area Method*. Coq user contribution. `http://dpt-info.u-strasbg.fr/~narboux/area_method.html`.

[22] Juan Paneque, Pedro Cobo, Josep Fortuny & Philippe R. Richard (2016): *Argumentative Effects of a Geometric Construction Tutorial System in Solving Problems of Proof*. In: *Proceedings of the 4th International Workshop on Theorem proving components for Educational software July 15, 2015 Washington, D.C., USA*, *CISUC Technical Reports* 2016-001, pp. 13–35. Available at `https://www.cisuc.uc.pt/ckfinder/userfiles/files/TR2016-01.pdf`.

[23] Pedro Quaresma (2011): *Thousands of Geometric Problems for Geometric Theorem Provers (TGTP)*. In Pascal Schreck, Julien Narboux & Jürgen Richter-Gebert, editors: *Automated Deduction in Geometry*, *Lecture Notes in Computer Science* 6877, Springer, pp. 169–181, doi:10.1007/978-3-642-25070-5_10.

[24] Pedro Quaresma (2017): *Towards an Intelligent and Dynamic Geometry Book*. *Mathematics in Computer Science* 11(3–4), pp. 427–437, doi:10.1007/s11786-017-0302-8.

[25] Pedro Quaresma & Nuno Baeta (2015): *Current Status of the I2GATP Common Format*. In Francisco Botana & Pedro Quaresma, editors: *Automated Deduction in Geometry: 10th International Workshop, ADG 2014, Coimbra, Portugal, July 9–1, 2014, Revised Selected Papers*, *Lecture Notes in Artificial Intelligence* 9201, Springer, pp. 119–128, doi:10.1007/978-3-319-21362-0_8.

[26] Pedro Quaresma, Vanda Santos, Pierluigi Graziani & Nuno Baeta (2019): *Taxonomies of geometric problems*. *Journal of Symbolic Computation*, (in press), doi:10.1016/j.jsc.2018.12.004.

[27] Philippe Richard, Pedro Cobo, Josep Fortuny & Markus Hohenwarter (2009): *Training teachers to manage problem-solving classes with computer support*. *Revista de Informática Aplicada / Journal of Applied Computing* 5(1), pp. 38–50, doi:10.13037/rasvol5n1.

[28] Mary Budd Rowe (1972): *Wait-Time and Rewards as Instructional Variables: Their Influence on Language, Logic, and Fate Control*. Technical Report, National Association for Research in Science Teaching. Available at `https://files.eric.ed.gov/fulltext/ED061103.pdf`.

[29] Robert J. Stahl (1994): *Using "Think-Time" and "Wait-Time" Skillfully in the Classroom*. Technical Report, ERIC Digest. Available at `http://files.eric.ed.gov/fulltext/ED370885.pdf`.

[30] Sana Stojanović, Vesna Pavlović & Predrag Janičić (2011): *A Coherent Logic Based Geometry Theorem Prover Capable of Producing Formal and Readable Proofs*. In Pascal Schreck, Julien Narboux & Jürgen

Richter-Gebert, editors: *Automated Deduction in Geometry: 8th International Workshop, ADG 2010, Munich, Germany, July 22-24, 2010, Revised Selected Papers*, *Lecture Notes in Artificial Intelligence* 6877, Springer, pp. 201–220, doi:10.1007/978-3-642-25070-5_12.

[31] Geoffrey Sutcliffe (2016): *The 8th IJCAR automated theorem proving system competition - CASC-J8*. *AI Communications* 29(5), pp. 607–619, doi:10.3233/AIC-160709.

[32] D. Wang (1995): *Reasoning about geometric problems using an elimination method*. In J. Pfalzgraf & D. Wang, editors: *Automated Pratical Reasoning*, Springer, New York, pp. 147–185, doi:10.1007/978-3-7091-6604-8_8.

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

[34] W.T. Wu (1984): *Automated Theorem Proving: After 25 Years*, chapter On the decision problem and the mechanization of theorem proving in elementary geometry, pp. 213–234. 29, American Mathematical Society, doi:10.1090/conm/029.

[35] Zheng Ye, Shang-Ching Chou & Xiao-Shan Gao (2010): *Visually Dynamic Presentation of Proofs in Plane Geometry: Part 1. Basic Features and the Manual Input Method*. *Journal of Automated Reasoning* 45(3), pp. 213–241, doi:10.1007/s10817-009-9162-5.

[36] Zheng Ye, Shang-Ching Chou & Xiao-Shan Gao (2010): *Visually Dynamic Presentation of Proofs in Plane Geometry: Part 2. Automated Generation of Visually Dynamic Presentations with the Full-Angle Method and the Deductive Database Method*. *Journal of Automated Reasoning* 45(3), pp. 243–266, doi:10.1007/s10817-009-9163-4.

[37] Zheng Ye, Shang-Ching Chou & Xiao-Shan Gao (2011): *An Introduction to Java Geometry Expert*. In Thomas Sturm & Christoph Zengler, editors: *Automated Deduction in Geometry*, *Lecture Notes in Computer Science* 6301, Springer Berlin Heidelberg, pp. 189–195, doi:10.1007/978-3-642-21046-4_10.

[38] Jing-Zhong Zhang, Shang-Ching Chou & Xiao-Shan Gao (1995): *Automated production of traditional proofs for theorems in Euclidean geometry: I. The Hilbert intersection point theorems*. *Annals of Mathematics and Artificial Intelligence* 13(1–2), pp. 109–137, doi:10.1007/BF01531326.