

# Towards a Geometry Automated Provers Competition\*

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

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

Zoltán Kovács  
The Private University College of Education of the Diocese of Linz  
Linz, Austria  
zoltan@geogebra.org

The geometry automated theorem proving area distinguishes itself by a large number of specific methods and implementations, different approaches (synthetic, algebraic, semi-synthetic) and different goals and applications (from research in the area of artificial intelligence to applications in education).

Apart from the usual measures of efficiency (e.g. CPU time), the possibility of visual and/or readable proofs is also an expected output against which the geometry automated theorem provers (GATP) should be measured.

The implementation of a competition between GATP would allow to create a test bench for GATP developers to improve the existing ones and to propose new ones. It would also allow to establish a ranking for GATP that could be used by “clients” (e.g. developers of educational e-learning systems) to choose the best implementation for a given intended use.

## 1 Introduction

The area of geometry automated theorem proving distinguishes itself by a large number of specific methods and implementations. Synthetic methods try to automate the traditional geometric proving processes [34, 38]; although being able to produce readable proofs, the so far proposed methods are very narrow-scoped and not efficient. The algebraic methods reduce the complexity of logical inferences by translating the geometric conjecture to an algebraic conjecture and then applying a given algebraic method. 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 and efficient. However, if eventually a proof record is produced, it will be a very complex algebraic proof [38]. In order to combine the geometric reasoning of synthetic methods and the efficiency of algebraic methods, some approaches, such as the area method and the the full-angle method, represent geometric knowledge in a form of expressions with respect to geometric invariants. These methods are broad-scoped, efficient and capable of producing geometric proofs [10, 11, 19].

When considering the geometric automated theorem provers (GATP), questions of applicability, e.g. in education, are very important. The improvement of existing implementations or the goals to be attained by new methods/implementations must take in consideration not only research goals but also the practical intended usefulness.

---

\*This work is financed by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2019

To be able to compare the different methods and implementations, a competition will have the virtue of pushing towards the standardization of the input language, the standardization of test sets, the direct comparability and the easier exchange of ideas and algorithmic techniques. The results of such a competition will also constitute a showcase, where potential users will look for the best GATP for their goals.

Towards a *Geometry Automated theorem provers System Competition (GASC)* many steps must be developed and combined. Quoting from TOOLympics 2019<sup>1</sup> [3, 4]:

1. How to assess adequacy of benchmark sets, and how to establish suitable input formats? And what is a suitable license for a benchmark collection?
2. How to execute the challenges (on-site vs. off-site, on controlled resources vs. on individual hardware, automatic vs. interactive, etc.)?
3. How to evaluate the results, e.g. in order to obtain a ranking?
4. How to ensure fairness in the evaluation, e.g. how to avoid bias in the benchmark sets, how to reliably measure execution times, and how to handle incorrect or incomplete results?
5. How to guarantee reproducibility of the results?
6. How to achieve and measure progress of the state of the art?
7. How to make the results and competing tools available so that they can be leveraged in subsequent events?

Some partial results are already available: a common language to state the geometric theorems [29], a comprehensive repository of geometric problems [28] and a set of measures of quality capable of assessing the GATPs in different classes [2, 31].

The ideas behind such a competition were presented in [2]. From the subsequent discussion, where a small set of six geometric problems was chosen, we progressed to an actual competition at ThEdu'19, GASC 0.1, run in a local computer, and where a set of GATPs competed over problems in the TGTP database. This paper is the result of all the discussions occurred there. For GASC 0.2 the complete set of problems in TGTP was used and the competition was conducted over the Internet, using a Web-server to run the competition and clients to check for the running of the competition and its final results.

*Overview of the paper.* The paper is organised as follows: first, in Section 2, some current GATP are presented. In Section 3 the repository of geometric problems is presented. In Section 4, the question related to the common format are discussed. In Section 5 the practical question about the implementation of the competition are discussed. In Section 6 the different measures of quality are discussed. Final conclusions are drawn and future work is foreseen in Section 7.

## 2 GATPs

For the preliminary run of *GASC*, during ThEdu'19,<sup>2</sup> the *GCLC* set of provers and some of the provers in *GeoGebra's portfolio prover* were selected [5, 19, 20]. The criteria used was: availability, reliability, and the possibility of running them in the command line (i.e. in a stand-alone fashion, outside a given computational tool where they could be integrated).

---

<sup>1</sup><https://tacas.info/toolympics.php>

<sup>2</sup>ThEdu'19, the 8th International Workshop on Theorem proving components for Educational software, 25 August 2019, Natal, Brazil

In future editions of *GASC* we hope that such set of provers can be enlarged as much as possible. A list of possible candidates is given by:

**GCLC** *GCLC* is a tool for visualizing objects and notions of geometry and other fields of mathematics, by generating figures and animations in the *gc* language. It has a built-in geometry theorem prover that can automatically prove a range of complex problems. The GATP module implements the *Area Method*, the *Wu's method* and the *Gröbner Basis Method* [18, 19].

The implemented GATP can be called from inside the *gcl-workbench* (Linux) or *WinGCLC* (MS-Windows) DGS tools, but can be also used in an independent way. Whenever called, in the command line it will produce (if successful) a rendering of the construction and a proof record, both to be processed by a  $\text{\LaTeX}$  compiler.

```
$ gclc GE00001.gcl                               Area Method (default method) [19];
$ gclc GE00001.gcl -w                             Wu's method [9];
$ gclc GE00001.gcl -g                             Gröbner bases method [9].
```

**OpenGeoProver** Open Library of Geometry Automatic Theorem Provers, *OpenGeoProver*.<sup>3</sup> It is an open source project, aiming to implement various geometry automated theorem provers. It can be used as a stand-alone tool but can also be integrated into other geometry tools, such as dynamic geometry software, e.g. work is being made to integrate *OpenGeoProver* with GeoGebra [25]. In its current state, *OpenGeoProver* implements the Wu's method. Some work has already been done to include implementations of the area method and the full-angle method [1].

```
$ ./runOGP GE00001.xml                           Wu's method [9];
```

this is a *bash* script that calls the OGP prover (*Java* bytecode).

**CoqAM** The formalisation of the area method using the proof assistant *Coq* was done by implementing the decision procedure as a *Coq* tactic and formalising all theorems needed by the method. The implementation guarantee the soundness of the method implementation, i.e., the proofs generated by the tactic are always correct [19, 23].

```
$ coqc GE00001.v > GE00001.errors                 Area Method in Coq [19, 23].
```

**GeoGebra's portfolio prover** GeoGebra has an embedded prover system that is capable of using multiple internal backends for proving theorems [20]. Its *Prove* and *ProveDetails* commands are the user level interface to formalize statements in the given syntax. They are considered as low-level commands because most users want to compare geometric objects directly by using the **Relation Tool** in GeoGebra, and by having an automated conjecture that is based on numerical checks, the low-level commands will also be issued by GeoGebra automatically.

The backends include Recio's exact check method [22], the Gröbner Basis Method, and *OpenGeoProver* can also be internally used to perform computations via Wu's method. For the Gröbner Basis Method it is possible to use an internal implementation of computing Gröbner bases via the *Giac* computer algebra system [21], or to use an external system that uses *Singular* [6, 7].

```
$ xvfb-run geogebra -prover=engine:Recio GE00001.ggb   Recio's exact check
method [22];
$ xvfb-run geogebra -prover=engine:Botana GE00001.ggb   Gröbner Basis
method [6, 7].
```

---

<sup>3</sup><https://github.com/opengeometryprover>

GeoGebra’s portfolio prover system automatically decides which backend should be used, but currently the best results can be obtained with the Gröbner Basis Method via *Giac* [24]. On the other hand, the selection of the backend can be fine-tuned by using command line options of GeoGebra in its desktop version. GeoGebra internally has a problem repository that is used for testing each backend on a daily basis—the results are published and updated at [prover-test.geogebra.org](http://prover-test.geogebra.org)<sup>4</sup> by a Jenkins system<sup>5</sup> installed at that server, providing a continuous checking of the results.

There are many other GATPs to be considered: *ArgoCLP*, a Coherent Logic Based Geometry Theorem Prover [34], *GEOTHER* [36], *Gex* [12, 13, 15], *JGEX* [39], *MMP* [14]; different formalizations in Coq: an automatic prover for projective geometry [8]; Gröbner basis method [26, 27]; Buchberger’s algorithm [17]; Wu’s method [16]. The first-order generic theorem provers must also be considered.

The challenge is to be able to incorporate them in GASC along any other GATP not listed above and/or any new system in current development.

### 3 Repositories of Geometric Problems

To test the GATP a test suite of problems must be created for each edition of the competition. The repository of problems *Thousands of Geometric problems for geometric Theorem Provers* (TGTP) could be used for such effect. TGTP is a Web-based repository of problems in geometry, with a significant size.<sup>6</sup> It also provides a supporting library to allow the use of the repository by different GATP [28].

The set of problems in consideration should also consider different axiom systems: neutral geometry; euclidean geometry; hyperbolic geometry; projective geometry; etc. Different types of conjectures should also be considered: constructive geometry; ruler and compass construction problems, conjectures involving inequalities.

After each edition of the GASC the (eventually) new problems would increase the TGTP repository.

### 4 GATPs Common Format

To be able to proceed, involving more GATPs, diversifying the axiom systems and the type of conjectures, a common format must be developed.

The I2GATP format is an extension of the I2G (Intergeo) common format aimed to support conjectures and proofs produced by geometric automatic theorem provers. The goal in building such a format is to provide a communication channel between different tools from the field of geometry, allowing the linking of such tools, as well as allowing the use of geometric knowledge kept in different repositories [29]. The TGTP repository and accompanying library of filters support the I2GATP.<sup>7</sup>

Having that (or other common format) filters from the common format to the new GATPs in the competition must be implemented just before the start of the competition.<sup>8</sup>

<sup>4</sup><https://prover-test.geogebra.org/job/GeoGebra-provertest/ws/test/scripts/benchmark/prover/html/all.html>

<sup>5</sup><https://github.com/jenkinsci/jenkins>

<sup>6</sup>v2.1.91—236 problems, <http://hilbert.mat.uc.pt/TGTP>

<sup>7</sup>Library of filters supporting the I2GATP common format <https://github.com/GeoTiles/libI2GATP>

<sup>8</sup>From GeoThms to GeoGebra: <https://github.com/kovzol/GeoThms2ggb>.

## 5 GASC 0.2 Competition

Using the TGTP database a set of 224 problems (geometric conjectures) were selected. The GCLC code and the Coq area method code were used. A filter from the GCLC code to GeoGebra code was implemented. In the future all the problems should be in a common format, with filters for all and each GATP in competition.

The first step toward a **Geometry Automated theorem provers Systems Competition (GASC)** was given at ThEdu'19, a presentation was made and a first trial, GASC 0.1, was conducted in a local computer (the second author laptop) using two scripts: one to launch the competition and follow it and another script to see the results in a never ending loop.

After ThEdu'19, GASC 0.2 was run incorporating all the comments received during the workshop, e.g. the TOOLympics reference [4]. The major difference between GASC 0.1 and GASC 0.2 is in the use of an Web server to support the competition.<sup>9</sup>

The server that supported GASC 0.2 was a Linux system, `Linux 4.9.0-2-amd64 #1 SMP Debian 4.9.18-1 (2017-03-30) x86_64 GNU/Linux`. The desktop computer motherboard is a Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz with 16GiB of RAM.

The possibility of continuing to run GASC on a dedicated server or to change to a specialized platform like StarExec [35] is an open question.

## 6 Results & Taxonomies

Apart the simple measure of speed of execution (CPU times), GATP should also be evaluated by other criteria, such as: readability of the proof produced and usability, e.g. in an educational setting.

How to measure the readability of proofs is still a research problem [2, 30, 31], Chou proposed a way to measure how difficult a formal proof is (using the area method) [9], de Bruijn also proposed a coefficient, the de Bruijn factor, 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 [37]. This is close to a Turing test for proofs: if a human cannot distinguish the proof generated automatically from a human proof, than it is readable.

Up to now this issue was not addressed. A first simple binary criteria, has a readable geometric proof or not, could be used to start.

The other criteria, the GATP usability in an educational setting, can be analysed in two different ways. The formal validation of a given conjecture, i.e. given a construction done using a DGS the possibility of having a formal validation of conjectures over that construction, or the use of the proof as a learning object by itself.

For the validation of conjectures the important factor is (again) the time, or more precisely the “wait-time”—periods of silence that followed teacher questions and students’ completed responses [32, 33]. The following classes of “GATP validation time” could be defined, in terms of time,  $t$ , taken by the GATP to answer [31]:

- good:  $t \leq 1.5s$ ;
- fair:  $1.5s < t \leq 3s$ ;
- poor:  $t > 3s$ .

---

<sup>9</sup><http://hilbert.mat.uc.pt/GASC/>

For the use of the proof as learning objects, we are back to the readability of GATP proofs, so, again, a binary choice between “maybe” or “not available” is, for now, the only possible outcome.

## 7 Future Work

All the research and technical issues about the competition, described above, must be solved/fixed.

The organization of the competition in the long term would require the support of the geometry automated deduction community: by entering the competition; by setting a problems committee that would choose the set of problems to be solved by the GATP, and maybe the more important point, by using its outcomes to their research and/or applications.

It is planned that a new zero-edition (0.3) will be implemented at ThEdu’20 (workshop at the International Joint Conference on Automated Reasoning, IJCAR 2020, June 29–July 5, 2020, Paris, France) and that the first edition of GASC would occur at the 13th International Workshop on Automated Deduction in Geometry, ADG 2020, Hagenberg, Austria, 13-15 July, 2020.

## References

- [1] Nuno Baeta & Pedro Quaresma (2013): *The full angle method on the OpenGeoProver*. In Christoph Lange, David Aspinall, Jacques Carette, James Davenport, Andrea Kohlhase, Michael Kohlhase, Paul Libbrecht, Pedro Quaresma, Florian Rabe, Petr Sojka, Iain Whiteside & Wolfgang Windsteiger, editors: *MathUI, OpenMath, PLMMS and ThEdu Workshops and Work in Progress at the Conference on Intelligent Computer Mathematics, CEUR Workshop Proceedings 1010*, Aachen. Available at <http://ceur-ws.org/Vol-1010/paper-08.pdf>.
- [2] Nuno Baeta & Pedro Quaresma (2019): *Towards Ranking Geometric Automated Theorem Provers*. In Pedro Quaresma & Walther Neuper, editors: *Proceedings 7th International Workshop on Theorem proving components for Educational software*, Oxford, United Kingdom, 18 July 2018, *Electronic Proceedings in Theoretical Computer Science 290*, Open Publishing Association, pp. 30–37, doi:10.4204/EPTCS.290.3.
- [3] Ezio Bartocci, Dirk Beyer, Paul E. Black, Grigory Fedyukovich, Hubert Garavel, Arnd Hartmanns, Marieke Huisman, Fabrice Kordon, Julian Nagele, Mihaela Sighireanu, Bernhard Steffen, Martin Suda, Geoff Sutcliffe, Tjark Weber & Akihisa Yamada (2019): *TOOLympics 2019: An Overview of Competitions in Formal Methods*. In Dirk Beyer, Marieke Huisman, Fabrice Kordon & Bernhard Steffen, editors: *Tools and Algorithms for the Construction and Analysis of Systems, Lecture Notes in Computer Science 11429*, Springer International Publishing, pp. 3–24, doi:10.1007/978-3-030-17502-3\_1.
- [4] Dirk Beyer, Marieke Huisman, Fabrice Kordon & Bernhard Steffen, editors (2019): *Tools and Algorithms for the Construction and Analysis of Systems: 25 Years of TACAS: TOOLympics. LNCS 11429*, Springer, doi:10.1007/978-3-030-17462-0.
- [5] 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.
- [6] Francisco Botana & Zoltán Kovács (2014): *A Singular web service for geometric computations*. *Annals of Mathematics and Artificial Intelligence*, pp. 1–12, doi:10.1007/s10472-014-9438-2.
- [7] Francisco Botana, Zoltán Kovács & Simon Weitzhofer (2012): *Implementing theorem proving in GeoGebra by using a Singular webservice*. In: *Proceedings EACA 2012, Libro de Resúmenes del XIII Encuentro de Álgebra Computacional y Aplicaciones*, Universidad de Alcalá, pp. 67–70. Available at [https://www.researchgate.net/publication/277475035\\_Implementing\\_Theorem\\_Proving\\_in\\_GeoGebra\\_by\\_Using\\_a\\_Singular\\_Webservice](https://www.researchgate.net/publication/277475035_Implementing_Theorem_Proving_in_GeoGebra_by_Using_a_Singular_Webservice).

- [8] David Braun, Nicolas Magaud & Pascal Schreck (2018): *Formalizing Some “Small” Finite Models of Projective Geometry in Coq*. In Jacques Fleuriot, Dongming Wang & Jacques Calmet, editors: *Artificial Intelligence and Symbolic Computation*, LNCS 11110, pp. 54–69. Springer, Cham. doi:10.1007/978-3-319-99957-9\_4.
- [9] Shang-Ching Chou, Xiao-Shan Gao & Jing-Zhong Zhang (1994): *Machine Proofs in Geometry*. World Scientific. doi:10.1142/2196.
- [10] 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(13), pp. 325–347, doi:10.1007/BF00283133.
- [11] 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(13), pp. 349–370, doi:10.1007/BF00283134.
- [12] Xiao-Shan Gao (1999): *Building Dynamic Mathematical Models with Geometry Expert, III. A Geometry Deductive Database*. In: *Proceedings Of ATCM’99*, ATCM Inc., USA, pp. 153–162.
- [13] Xiao-Shan Gao, C. C. Zhu & Y. Huang (1998.): *Building Dynamic Mathematical Models with Geometry Expert, II. Linkage*. In Z. B. Li, editor: *Proceedings of the Third Asian Symposium on Computer Mathematics*, LanZhou University Press, pp. 15–22.
- [14] Xiao-Shan Gao & Qiang Lin (2004): *MMP/Geometer - A Software Package for Automated Geometric Reasoning*. *Lecture Notes in Computer Science* 2930, pp. 44–66. doi:10.1007/978-3-540-24616-9\_4.
- [15] Xiao-Shan Gao, C. C. Zhu & Y. Huang (1998): *Building Dynamic Visual and Logic Models with Geometry Expert I*. In W.C. Yang, editor: *Proceedings of the Third Asian Technology Conference in Mathematics*, Springer, pp. 216–226.
- [16] Jean-David Génevaux, Julien Narboux & Pascal Schreck (2011): Formalization of Wu’s Simple Method in Coq. In Jean-Pierre Jouannaud & Zhong Shao, editors: *Certified Programs and Proofs*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 71–86. doi:10.1007/978-3-642-25379-9\_8.
- [17] Benjamin Grégoire, Loïc Pottier & Laurent Théry (2011): Proof Certificates for Algebra and Their Application to Automatic Geometry Theorem Proving. In Thomas Sturm & Christoph Zengler, editors: *Automated Deduction in Geometry*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 42–59. doi:10.1007/978-3-642-21046-4\_3.
- [18] Predrag Janičić (2006): GCLC — A Tool for Constructive Euclidean Geometry and More Than That. In Andrés Iglesias & Nobuki Takayama, editors: *Mathematical Software - ICMS 2006*, *Lecture Notes in Computer Science* 4151, Springer, pp. 58–73, doi:10.1007/11832225\_6.
- [19] 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.
- [20] Zoltán Kovács (2015): The Relation Tool in GeoGebra 5. In Francisco Botana & Pedro Quaresma, editors: *Automated Deduction in Geometry*, *Lecture Notes in Computer Science* 9201, Springer, pp. 53–71, doi:10.1007/978-3-319-21362-0\_4.
- [21] Zoltán Kovács & Bernard Parisse (2015): Giac and GeoGebra – Improved Gröbner Basis Computations. In Jaime Gutierrez, Josef Schicho & Martin Weimann, editors: *Computer Algebra and Polynomials*, *Lecture Notes in Computer Science*, Springer, pp. 126–138, doi:10.1007/978-3-319-15081-9\_7.
- [22] Zoltán Kovács, Tomás Recio & Simon Weitzhofer (2012): Implementing theorem proving in GeoGebra by using exact check of a statement in a bounded number of test cases. In: *Proceedings EACA 2012, Libro de Resúmenes del XIII Encuentro de Álgebra Computacional y Aplicaciones*, Universidad de Alcalá, pp. 123–126. Available at [https://www.researchgate.net/publication/265059624\\_Implementing\\_Theorem\\_Proving\\_in\\_GeoGebra\\_by\\_Exact\\_Check\\_of\\_a\\_Statement\\_in\\_a\\_Bounded\\_Number\\_of\\_Test\\_Cases](https://www.researchgate.net/publication/265059624_Implementing_Theorem_Proving_in_GeoGebra_by_Exact_Check_of_a_Statement_in_a_Bounded_Number_of_Test_Cases).
- [23] Julien Narboux (2009): Formalization of the Area Method. Coq user contribution. [http://dpt-info.u-strasbg.fr/~narboux/area\\_method.html](http://dpt-info.u-strasbg.fr/~narboux/area_method.html).

- [24] Mladen Nikolić, Vesna Marinković, Zoltán Kovács & Predrag Janičić (2018): Portfolio theorem proving and prover runtime prediction for geometry. *Annals of Mathematics and Artificial Intelligence*, pp. 1–28, doi:10.1007/s10472-018-9598-6.
- [25] Ivan Petrović & Predrag Janičić (2012): Integration of OpenGeoProver with GeoGebra. Available at <http://argo.matf.bg.ac.rs/events/2012/fatpa2012/slides/IvanPetrovic.pdf>.
- [26] Loïc Pottier & Laurent Théry (1998): gbcoq. <http://www-sop.inria.fr/croap/CFC/Gbcoq.html>.
- [27] Loïc Pottier (2008): Connecting Gröbner Bases Programs with Coq to do Proofs in Algebra, Geometry and Arithmetics. *Knowledge Exchange: Automated Provers and Proof Assistants 418*. Available at <http://arxiv.org/abs/1007.3615>.
- [28] 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.
- [29] Pedro Quaresma & Nuno Baeta (2015): Current Status of the I2GATP Common Format. In Francisco Botana & Pedro Quaresma, editors: *Automated Deduction in Geometry, Lecture Notes in Artificial Intelligence 9201*, Springer, pp. 119–128. doi:10.1007/978-3-319-21362-0\_8.
- [30] Pedro Quaresma & Vanda Santos (2019): Proof Technology in Mathematics Research and Teaching, chapter Computer-generated geometry proofs in a learning context. Springer. doi:10.1007/978-3-030-28483-1.
- [31] Pedro Quaresma, Vanda Santos, Pierluigi Graziani & Nuno Baeta (2019): Taxonomies of geometric problems. *Journal of Symbolic Computation*. 97, pp. 31–55. Elsevier. doi:10.1016/j.jsc.2018.12.004.
- [32] 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>.
- [33] 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>.
- [34] 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, Lecture Notes in Computer Science 6877*, Springer Berlin Heidelberg, pp. 201–220, doi:10.1007/978-3-642-25070-5\_12.
- [35] Aaron Stump, Geoff Sutcliffe & Cesare Tinelli (2014): StarExec: A Cross-Community Infrastructure for Logic Solving. In Stéphane Demri, Deepak Kapur & Christoph Weidenbach, editors: *Automated Reasoning, Springer International Publishing, Cham*, pp. 367–373. doi:10.1007/978-3-319-08587-6\_28.
- [36] Dongming Wang (2004): GEOTHER 1.1: Handling and Proving Geometric Theorems Automatically. In Franz Winkler, editor: *Automated Deduction in Geometry, Lecture Notes in Computer Science 2930*, Springer, Berlin, Heidelberg, pp. 194–215, doi:10.1007/978-3-540-24616-9\_12.
- [37] 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.
- [38] Wen-Tsun 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. *Contemporary Mathematics 29*, American Mathematical Society, doi:10.1090/conm/029/12.
- [39] 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.