Thousands of Geometric Problems for Geometric Theorem Provers (TGTP)

Pedro Quaresma

CISUC, Department of Mathematics, University of Coimbra
3001-454 Coimbra, Portugal
pedro@mat.uc.pt

Abstract. Thousands of Geometric problems for geometric Theorem Provers (*TGTP*) is a Web-based library of problems in geometry.

The principal motivation in building *TGTP* is to create an appropriate context for testing and evaluating geometric automated theorem proving systems (GATP). For that purpose *TGTP* provides a centralised common library of geometric problems with an already significant size but aiming to became large enough to ensure meaningful system evaluations and comparisons. *TGTP* provides also a workbench were it is possible to test any given geometric conjecture.

TGTP is independent of any given GATP. For each problem the code for each GATP (whenever available) is kept in the library. A common format for geometric conjectures, extending the i2g format, is being developed. This common format, plus a list of converters, one for each GATP, will allow to test all the GATPs with all the problems in the library.

TGTP is well structured, documented and with a powerful querying mechanism, allowing an easy access to the information. All information in the library, and also the supporting formats and tools are freely available.

TGTP aims, in a similar spirit of *TPTP* and other libraries, to provide the automated reasoning in geometry community with a comprehensive and easily accessible library of GATP test problems. The development of *TGTP* problem library is an ongoing project.

Keywords: Library of problems in geometry, Geometric Automated Theorem Proving.

1 Introduction

Automated theorem provers, applications, and libraries of problems are often developed separately. In some cases, joint efforts of many of researchers led to standards such as *DIMACS* (for propositional logic) [6] and *SMT* (for satisfiability modulo theory) [1] and libraries of problems such as *SATLIB* (for propositional logic) [10], *TPTP*[1] (for predicate logic) [21], *SMT-lib* (for satisfiability modulo theory) [1] etc. Such efforts, standards, and libraries are fruitful for easier exchange of problems, ideas, and even program code. However, this is often

[1] http://www.cs.miami.edu/~tptp

P. Schreck, J. Narboux, and J. Richter-Gebert (Eds.): ADG 2010, LNAI 6877, pp. 169–181, 2011.

very demanding and there are not many systems smoothly integrating libraries of problems and theorem provers.

There are also several systems integrating dynamic geometry software (DGS), GATPs, and a set of examples. For example: *Java Geometry Expert*[2] (*JGEX*) is a system that combines dynamic geometry, automated geometry theorem proving and visual dynamic presentation of proofs. It contains a large set of examples; *GEOTHER*[3] is an environment for manipulating and proving geometric theorems implemented in Maple. It contains a collection of theorems in both elementary and differential geometry [22]; *Ludi Geometrici*[4] has a vast library of problems in the area of classical constructive (ruler and compass only) Euclidean geometry. It does not provide a GATP so no formal proofs are provided; *GeoThms*[5] is a Web workbench in the field of constructive problems in Euclidean geometry. It links DGSs and GATPs and contains a large library of geometry problems [19]. Many of the DGSs, e.g. *GeoGebra*[6] [8], *Cabri*[7], *Cinderella*[8][15,20], etc, DGSs/GATPs, e.g. *GCLC* [12], *GeoView* [2], *GeoProof* [16], *Geometry Explorer* [23], *MMP/Geometer*[9] [9],*GEX* [9], *Discover* [3], and also GATPs like *Theorema* [4] come with a (some times large) set of examples. However none of these systems try to provide a common platform for meaningful system evaluations and comparisons.

In the rest of this paper we present Thousands of Geometric problems for geometric Theorem Provers (*TGTP*[10]) which is a Web-based library of GATP test problems. It aims to become a comprehensive common library of problems with a significant size and unambiguous reference mechanism, easily accessible to all researchers in the automated reasoning in geometry community. *TGTP* tries to address all relevant issues. In particular:

- is Web-based and is thus easily available to the research community;
- is easy to use;
- aims to provide a common format to conjectures in geometry;
- tries to cover the different forms of automated proving in geometry, e.g. synthetic proofs and algebraic proofs;
- aims to become large enough for statistically significant testing. In its current version it contains already over 180 problems;
- aims to become a comprehensive, up-to-date library;
- is independent of any particular GATP system;
- is well structured and documented. This allows effective and efficient use of the library;

[2] `http://www.cs.wichita.edu/~ye/`;
[3] `http://www-calfor.lip6.fr/~wang/GEOTHER/`
[4] `http://www.polarprof.org/geometriagon/`
[5] `http://hilbert.mat.uc.pt/GeoThms/`
[6] `http://www.geogebra.org/cms/`
[7] `http://www.cabri.com/`
[8] `http://www.cinderella.de`
[9] `http://www.mmrc.iss.ac.cn/~xgao/software.html`
[10] `http://hilbert.mat.uc.pt/TGTP`

- documents each problem. This contributes to the unambiguous identification of each problem;
- provides query mechanisms;
- provides a mechanism for adding new problems;
- provides a workbench for an easy testing of any given conjecture.

Paper Overview. Section 2 describes the *TGTP* system, its realm, the Web-interface, the structure of the information, the queries, the performance information. Section 3 talks about a common format for geometric conjectures. Section 4 discusses further work, and in Section 5 some final conclusions are drawn.

2 TGTP

Thousands of Geometric problems for geometric Theorem Provers (TGTP) is a Web-based library of geometric problems for testing and evaluating geometric automated theorem proving systems.

2.1 Realm

TGTP is a library of problems (conjectures) in geometry for GATP systems evaluation. *TGTP* aims to supply the automatic reasoning in geometry community with a comprehensive library of GATPs problems.

The *TGTP* library is independent of any GATP system, for each problem generic information is kept (see Section 2.3 for details) and, connected to this, the code for the different GATPs that are already associated with the problem.

A common XML-format is being developed based in the author's previous experience [18] and in the i2g common file format [7], extending this last format, allowing it to cope with conjectures. From this common format converters to GATP specific formats will be written, which can be used to provide the GATPs code whenever a specific realisation was not provided.

As said above *TGTP* stores, for each problem, some generic information, namely the name of the problem, a short textual information, a formal statement of the conjecture, a set of keywords and bibliographic references (some of this fields are optional), this linked with powerful query mechanisms allows keeping the list of problems coherent, avoiding duplications (see Section 2.3 for details).

The goal for building *TGTP* is, in a similar spirit of *TPTP* and other libraries, to provide the GATP community with a centralised problem collection with an easy access to all researchers. The *TGTP* aims to become a comprehensive up-to-date library of problems for the GATPs testing and evaluation.

2.2 The Web Interface

The *TGTP*'s Web interface aims to fulfil the goal of an easy availability of all the information to the GATP community. It is structured in only three levels (see Fig. 1), two, if we do not consider the entry level: a first level for login and also to browse some generic info about the system (HELP) and a second level (after the login) divided in four sections plus a LOGOUT option.

There are three different type of *TGTP*'s users: anonymous/regular users, contributors and the administrator. The administrator has access to a simple interface that allows to see logging information and to do some administrative duties.

The anonymous/regular user has access to the "public" interface. All the access is given in terms of "see but do not touch" mode. Exception to this is the WORKBENCH ,where any user can test the problems with the already installed provers. A personal scrapbook, i.e. a list of problems, is available. The anonymous users will share a common list, the other (registered) users will have his/her own list. This type of user has full access to the information and to the downloadable materials.

Contributors will have, in addition to all the regular users' features, the ability to add new problems, i.e., in the section "Problems List" the contributors will have the possibility of submit new problems and/or update the existing ones (see markers (r)egular and (c)ontributer in Fig. 1)

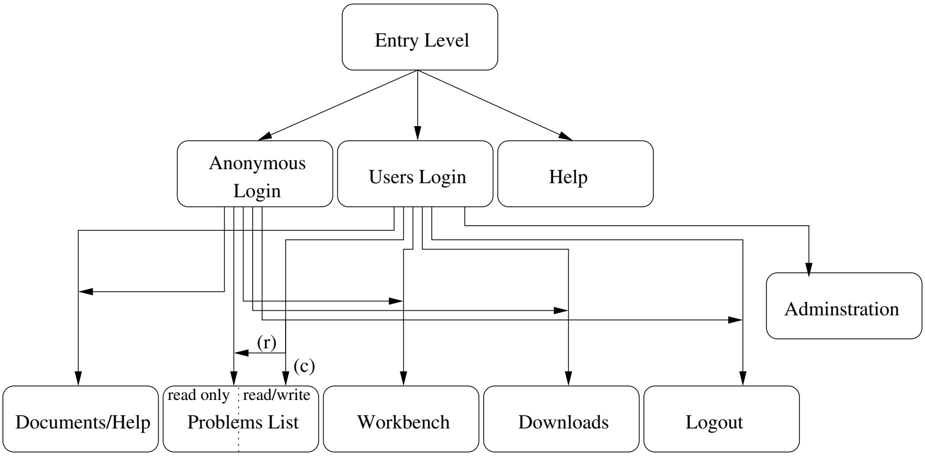

Fig. 1. Structure of the Web-page

The contributors can also produce a new set of evaluation data, i.e. a new set of performance values for the different GATPs when run over the *TGTP* set of problems. For instance, after the introduction of a new set of problems.

The *TGTP* shares with the *GeoThms* system the list of users.

The interface is divided in six main sections (see Fig. 1). The *Administration* is the section reserved for administrative duties. The *Logout* section is for a well-behaved exit, closing the Web-session and registering some information about the time spent by the user in the system.

The *Documents/Help* section contains documents, for instance, a list of bibliographic references about GATPs (in BibTeX format) cited, or not, in the problems; a list of provers and a list of authors with information about GATPs and its authors. It will also contain information regarding the use of *TGTP*: manuals, frequently asked question list, *How-To*s.

This section contains also the performance information regarding the GATPs and the list of problems: number of attempts, number of proof attempts succeeded, i.e. the GATP has reached a conclusion within the time limit of 600s; the percentage of success, and information of the CPU time spent in the proofs, the minimum time, the maximum time, and the average time. The information of each individual proof attempt is also displayed (see Section 2.5).

Also in this section is a link to the TGTP's Forum, a place where TGTP's users can freely exchange information.

The *Problem List* section contains the list of all problems introduced up to the present day. It is presented in a concise form: a list of 10 (or 20, or 50, or all) lines with the unique identification problem, the name of the problem, a short description (if present), and the number of proofs succeeded and the number of proof attempts. Each line contains also a link to another Web-page where all the info about the problem is presented.

For each problem it is possible to get all the details about it, and its proofs. From this page it is possible to download the information about the problem in textual form for an easy reading: its identification name, the submission date, its name, a short description and a formal statement (in LaTeX format), a list of keywords and for each proof attempt its status, the GATP used, and the GATP code.

The contributors can update/change the info on every existing problem in the database. They have also the capability to add new problems, the insertion of new problems is safeguarded with a validation step where a search for similar problems, already in the database, is done.

It will also be possible to submit a list of new problems for a bulk insertion into the database. The automatic processing of the list will be done with the help of a given XML-format (see Appendix A).

It is also possible to query the database to look for a problem or a set of related problems (see Section 2.4).

In the *Workbench* section it is possible to test conjectures with the "in-house" GATPs. A user (of any type) will have a simple Web-editor to write the conjecture he/she wants to submit to one of the GATPs that are available in the server, for now GCLCprover [13] and CoqAM [16]. The GATPs are called with a 600s time limit and after a successful run, or after 600s, the results of the proof are made available.

The problems to be submit can be: new problems, written by the user; existing problems, selected from *TGTP* list of problems, or from the personal scrapbook. The scrapbook is unique to every user.

The *Downloads* section is the place where it is possible to download documents related to the *TGTP* database itself and to GATP's codes listing.

The *TGTP* database can be, with the exception of the tables with the information of the *TGTP* users, downloaded in full, i.e. it is possible to download a file with the result of a "mysqldump" command [17]. It is also possible to download the entity-relationship diagram that describes the database (see Fig.2 for a condensed version of the ERD).

From this section the GATP's codes listings are also available, i.e. a text file with all the codes in the database related to any given GATP. This file is a simple text file with a simple separator between each problem's code. This lists of problems is also available in a compressed file containing the list of problems in XML format (see Appendix A) for an easy automatic parsing.

2.3 The List of Problems

The information is organised in five different aggregations (see Fig. 2). The aggregations *Conjectures* and *Proofs* are the core of *TGTP*. In *Conjectures* we have the list of all problems and in *Proofs* we have, for each problem, all its proofs attempts.

The *Users* aggregation is used to control the access of registered (and anonymous) users to *TGTP* and to keep information about the user's login history. The workbench is connected to this section by the `CodeTmpProver` table.

The aggregations *Measures of Efficiency* and *Statistics* (a more correct name should be *Performance Information*) have all the details about performance information. In *Statistics* a snapshot of all the measures of efficiency, at a given time, is kept. The purpose of this information is to keep an historical record of the *TGTP* status allowing an evaluation of the problems/GATPs/TGTP development along the years.

The `TGTP` table is used to keep track of different (majors) versions of *TGTP*.

Since *TGTP* shares with the *GeoThms* system the database of problems we can also have, for many of the conjectures but not necessarily for all, the DGS's code for the geometric construction. The DGS constructions are only available within the *GeoThms* system.

2.4 Queries

The list of problems can be queried in two ways: a simple query using MySQL's regular expressions and a more powerful using the full-text search of MySQL [17]. The first one is done over the `name` attribute of the table `Conjectures` after the user has provided a word to be searched. This word will be matched against any of the words in the list of words that constitute the conjectures names. The second one is done over the attributes `name, description, shortDescription` and `keywords` of the table `Conjectures` and allows, for a given input sentence, to get the list of most similar sentences in either of the these attributes.

2.5 Performance Information

The *TGTP* database contains now (2011/06/17) 180 problems and contains results of proof attempts from two GATP: CoqAM [16], and GCLCprover [13],

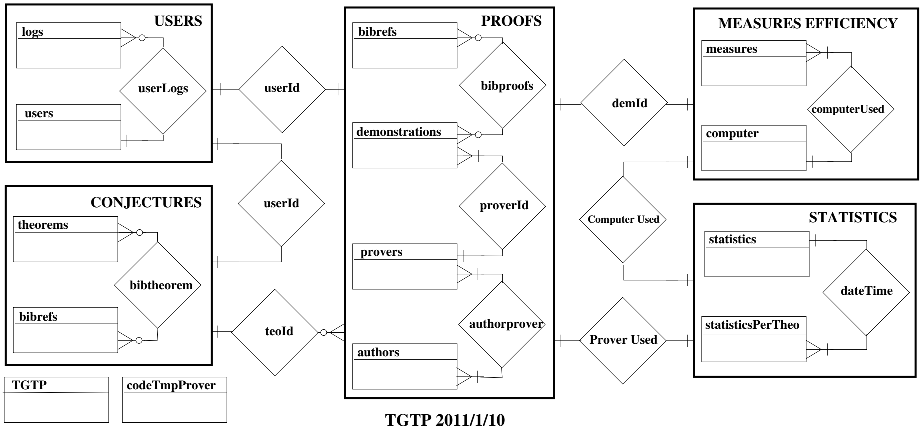

Fig. 2. Structure of the data base (E-R diagram)

covering the methods: Wu's method [24], Gröbner basis method [14] and area method [11].

A new set of performance values is taken whenever a major change in TGTP database occurs: a increase in the number of problems; a change in the computer that is used to run the GATPs; a inclusion of a new GATP or a change in version of an existing GATP. That is, all the data in the `measures` table is collected and tagged with the current date and saved in the table `statistics`. If needed, new measures of efficiency are taken, e.g. a new computer will imply a new set of measures for all the problems times all the GATPs; a new GATP, or a new version of an existing one, will imply a set of measures for all the problems with that particular GATP. For new problems all the GATPs (whenever applicable) are executed and the values added. A new version of *TGTP* is only relevant for this issue if the change would imply new, hopefully improved, codes for the problems, e.g. new converters, or a new version of the common format.

These snapshots of the information contained in *TGTP* and the fact that any kind of change: a new problem, a new proof to an existing problem; a new GATP, or a change in version of an existing GATP, etc, will add to the existing information (not update the information but to add new one) will allow to trace the evolution of a given GATP (through its changes of versions), or of a given problem, or the *TGTP* system itself.

The values are taken per proof attempt (see Table 1), that is, for each pair of problem and GATP's code, the performance values of that attempt are saved in the `measures` table. All the proofs attempts have a time limit of 600s after which the process is killed by the operating system.

The proof status are : "Proved"; "Disproved"; "Failed to prove the conjecture"; "Time-out: failed to prove the conjecture", when the process is killed before it reaches and end; "Maximal number of proof steps reached: failed to prove the conjecture", a limit that some GATPs (for example GCLC AM), have themselves; “The conjecture out of scope of the prover”, whenever the GATP could not deal with the problem, e.g. the provers based in the area method have a limited range of problems that they can deal with (see [11] for details). The correspondent numeric codes range from 1 to 6.

Table 1. Results of Proof Attempts (fragment)

	Coq (AM)		GCLC (AM)		GCLC (WM)		GCLC (GBM)	
TheoId	status	time	status	time	status	time	status	time
GEO0230			4	600.021	4	1.468	4	605.362
GEO0231	1	17.89	3	0	2	0.004	3	0.224
GEO0232			3	0.024	2	0	3	0.004
GEO0233			3	0.252	1	0.044	1	1.392
GEO0234	1	1.07	1	0	1	0	1	0
GEO0235	4	600.44	1	1.4	2	0.008	3	0.004
GEO0236	4	600.29	4	600.17	2	1.668	1	5.22
GEO0237	4	600.6	3	0.788	1	0.048	4	599.169
GEO0238	4	601.27	1	0.032	1	0.024	1	0.092
GEO0239			1	0.004	1	0.008	4	609.362

Apart from this, per problem results, some overall values are also collected (see Table 2). For each GATP the following measures are taken: the number of proof attempts, i.e. the number of code entries contained in the database; the number of times the GATP succeeded in proving (or disproving); the percentage of success; and some measures of CPU times: the minimum time needed, the maximum time needed, and the average time. This last values are taken only for those cases where the time-out limit was not reached.

The script used to run the GATPs, imposing a time limit, and getting the CPU time used by the GATPs, is this *bash script*:

```
#!/bin/bash
ulimit -t $1
/usr/bin/time --output=$2 -f "CPU time in seconds: %e" $3 $4 > $5
```

where `ulimit` and `time` are Linux tools to impose a time limit and to get the CPU time spent by a given process respectively. The arguments of the script are: the time limit (600s); the name of the file where the CPU time will be written; the name of the GATP; the argument (code) to the GATP; and the file that will receive (by a Linux redirection) all the output of the GATP.

After each run, a set of other scripts will parse the resulting files getting the desired results.

3 Common File Format for Conjectures

In [18] an XML-suite for constructive descriptions of geometrical figures and geometrical proofs is described. This format is used in the *GeoThms* system to

Table 2. Overall Results

	attempts	succeeded	%of success	min	max	avg
Coq (AM)	76	68	0.89	0.73	213.71	17.698
GCLC (AM)	123	62	0.5	0	360.235	9.194
GCLC (WM)	96	88	0.92	0	6.404	0.422
GCLC (GBM)	96	56	0.58	0	112.319	5.393

provide a common format for its list of problems, and where the conversion of this format to the DGSs/GATPs format is done via XSLT files.

Since then, the i2g common file format of the Intergeo consortium was specified, which is a file format designed to describe any construction made with the help a DGS [7].

Having this in mind we decided to adopt the i2g format and to extend it with an XML-based format for geometrical proofs (from our previous work). As said in [7] the Content Dictionaries [5] of OpenMath[11] can be used to define a new set of symbols, to describe geometric conjectures, and in this way to enrich the expressive power of the i2g common file format (see Appendix B for details).

We intend to support the automatic conversion from this new extended common format to all the GATPs formats available in the *TGTP* system.

4 Future Work

The *TGTP* project is, and it will always be, an ongoing project. New problems should be added to the existing list of problems, new GATPs, or new versions of existing GATPs should be considered.

Apart from this long term commitment, there are short/medium term improvements to be done: the common format for GATPs and the corresponding converters; direct conversions between the different GATPs (e.g. the Coq AM and GCLCprover) for an optimized comparison between GATPs; improvements in the performance information, namely the inclusion of graphical outputs for a better reading; improvements in the documentation and the Web-page.

5 Conclusions

In the *GeoThms* system the author of this article and Predrag Janičić already addressed some of the issues that are now being laid down for *TGTP*, namely the XML common format, and the list of problems. Where the *GeoThms* goal is to have a publicly accessible and widely used Internet based framework for constructive geometry with a strong integration of DGSs, GATPs and a library of problems, the *TGTP* goal is to provide the GATP community with a centralised collection of problems, independent of any particular GATP system.

The development of *TGTP* problem library is an ongoing project, aiming to provide all of the desired properties described above.

[11] `http://www.openmath.org/`

A List of Problems XML Format

The lists of problems (for each GATP) are available in files written in a simple XML format for an easy automatic parsing. This format is used for the bulk automatic insert of a given list of problems in the database, but it is also used to assemble a file with all the conjectures in *TGTP*. This file is accessible to download in the Web-page.

The XML format has the necessary tags to describe any given problem (an load it into the database). The tags are self-explanatory, the example below describes the format. The author of this article is open to any suggestions/improvements to this format that the readers might be willing to suggest.

```
<results>
 <gatpid>
  GATP id
 </gatpid>
 <result>
  <userid>
   Contributor Id (mandatory)
  </userid>
  <theoid>
    Theorem Id (output file, in the input file it will be ignored)
  </theoid>
  <theoname>
   Theorem Name (mandatory)
  </theoname>
  <description>
   Theorem statement in LaTeX format (optional)
  </description>
  <shortDescription>
   Theorem statement in text format (optional, but highly desirable)
  </shortDescription>
  <keywords>
   <keyword>
    keyword (list of keywords, optional, but highly desirable)
   </keyword>
   ...
  </keywords>
  <biblist>
   <bibitem>
    Bibliographic entry, in BibTeX format (optional)
   </bibitem>
   ...
  </biblist>
  <DGSid>
   <n> - the DGS id number (optional)
  </DGSid>
  <figcode>
   DGS code for the rendering of the Geometric Construction (optional)
  </figcode>
  <GATPid>
   <n> - the GATP id
  </GATPid>
  <proofscode>
   GATPs code
  </proofscode>
 </result>
 ...
</results>
```

The `biblist` and `keywords` lists may be empty.

The `theoid` tag is only meaningful when the XML file was generated by the *TGTP* system. If provided in the input file it will be ignored, the system will provide a unique identifier for each new problem disregarding any given value.

The `DGSid`s are: (0,i2g),(2, Eukleides - 1.0.2), (3,GCLC - 9.00), (4,GeoGebra - 3.2.0.0). The `GATPid`s are: (0,i2gGATP), (2,GCLC Area Method - 9.00), (4,COQ Area Method - 1.0), (5,GCLC Wu's Method - 9.00) and (6,GCLC Gröbner Basis Method - 9.00).

B The Common Format for GATPs

The proof methods considered in *TGTP* for now are: the area method, the Wu's method and the Gröbner Basis method. Having that in mind, we begin to introduce the symbols needed to support those methods. We have to consider algebraic polynomials, the area method quantities, and the geometric conjectures:

Algebraic Polynomials. The symbols needed for this can be imported from OpenMath Polynomial CD Group "polygrp"[12] which in turn use the symbols for arithmetic operators from other CDs (e.g. the "arith1" CD).

Area Method Symbols. The Area Method introduce the *ratio of directed segments*, the *signed area* and the *Pythagoras difference* of triangles and rectangles. Given that we will need to introduce:

```
sratio, signed_area3, signed_area4, pythagoras_difference3,
pythagoras_difference4
```

these symbols will be applied to points, the axiomatic elements of the area method, which are in the i2g CD.

The area method needs also the symbol for equality and the operators of a field $(F, +, \cdot, 0, 1)$ of characteristic different from 2, these symbols can be found in the CDs for arithmetic operators.

Symbols to express the *non-degeneracy conditions* [11] are also required.

[12] `http://www.openmath.org/cd/`

Geometric Conjectures. Introducing geometric conjectures we need the symbols for expressing conjectures, e.g. identical, collinear, perpendicular, parallel, midpoint, etc.

And also the symbols for the proof itself: conjecture, prove, lemmas.

An example of a file in this format is given below:

```
<conjecture>
  <equality>
   <expression>
    <signed_area3>
     <point>P</point>
     <point>Q</point>
     <point>R</point>
    </signed_area3>
   </expression>
   <expression>
    <number>0.000000</number>
   </expression>
  </equality>
 </prove>
</conjecture>
```

This is an ongoing project, any help will be welcome.

References

1. Barrett, C., Stump, A., Tinelli, C.: The SMT-LIB Standard: Version 2.0. In: Gupta, A., Kroening, D. (eds.) Proceedings of the 8th International Workshop on Satisfiability Modulo Theories, Edinburgh, England (2010)
2. Bertot, Y., Guilhot, F., Pottier, L.: Visualizing geometrical statements with GeoView. Electronic Notes in Theoretical Computer Science 103, 49–65 (2004)
3. Botana, F., Valcarce, J.L.: A dynamic-symbolic interface for geometric theorem discovery. Computers and Education 38, 21–35 (2002)
4. Buchberger, B., Craciun, A., Jebelean, T., Kovacs, L., Kutsia, T., Nakagawa, K., Piroi, F., Popov, N., Robu, J., Rosenkranz, M., Windsteiger, W.: Theorema: Towards computer-aided mathematical theory exploration. Journal of Applied Logic 4(4), 470–504 (2006)
5. Davenport, J.H.: On writing OpenMath content dictionaries. SIGSAM Bulletin 34(2), 12–15 (2000)
6. DIMACS: Satisfiability suggested format (May 1993), ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/doc/
7. Egido, S., Hendriks, M., Kreis, Y., Kortenkamp, U., Marquès, D.: i2g Common File Format Final Version. Tech. Rep. D3.10, The Intergeo Consortium (2010)
8. Fuchs, K., Hohenwarter, M.: Combination of dynamic geometry, algebra and calculus in the software system geogebra. In: Computer Algebra Systems and Dynamic Geometry Systems in Mathematics Teaching Conference 2004, pp. 128–133. Pécs, Hungary (2004)

9. Gao, X.-S., Lin, Q.: MMP/Geometer - a Software Package for Automated Geometric Reasoning. In: Winkler, F. (ed.) ADG 2002. LNCS (LNAI), vol. 2930, pp. 44–66. Springer, Heidelberg (2004)
10. Hoos, H., Stützle, T.: SATLIB: An online resource for research on SAT. In: Gent, I.P., Maaren, H.V., Walsh, T. (eds.) SAT 2000, pp. 283–292. IOS Press, Amsterdam (2000)
11. Janičić, P., Narboux, J., Quaresma, P.: The Area Method: a recapitulation. Journal of Automated Reasoning 7 (to appear, 2011), doi:10.1007/s10817-010-9209-7
12. Janičić, P.: GCLC — A Tool for Constructive Euclidean Geometry and More Than That. In: Iglesias, A., Takayama, N. (eds.) ICMS 2006. LNCS, vol. 4151, pp. 58–73. Springer, Heidelberg (2006)
13. Janičić, P., Quaresma, P.: System Description: GCLCprover + Geothms. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 145–150. Springer, Heidelberg (2006)
14. Kapur, D.: Using Gröbner bases to reason about geometry problems. Journal of Symbolic Computation 2(4), 399–408 (1986)
15. Kortenkamp, U., Richter-Gebert, J.: Using automatic theorem proving to improve the usability of geometry software. In: Procedings of the Mathematical User-Interfaces Workshop 2004 (2004)
16. Narboux, J.: A graphical user interface for formal proofs in geometry. Journal of Automated Reasoning 39, 161–180 (2007)
17. Oracle: MySQL 5.5 Reference Manual, 5.5 edn. (January 2011), revision: 24956
18. Quaresma, P., Janičić, P., Tomašević, J., Vujošević-Janičić, M., Tošić, D.: XML-Bases Format for Descriptions of Geometric Constructions and Proofs. In: Communicating Mathematics in The Digital Era, pp. 183–197. A. K. Peters, Ltd. (2008)
19. Quaresma, P., Janičić, P.: Geothms – a Web System for Euclidean Constructive Geometry. Electronic Notes in Theoretical Computer Science 174(2), 35–48 (2007)
20. Richter-Gebert, J., Kortenkamp, U.: The Interactive Geometry Software Cinderella. Springer, Heidelberg (1999)
21. Sutcliffe, G.: The TPTP problem library and associated infrastructure. Jounal of Automated Reasoning 43(4), 337–362 (2009)
22. Wang, D.: GEOTHER 1.1: Handling and Proving Geometric Theorems Automatically. In: Winkler, F. (ed.) ADG 2002. LNCS (LNAI), vol. 2930, pp. 194–215. Springer, Heidelberg (2004)
23. Wilson, S., Fleuriot, J.: Combining dynamic geometry, automated geometry theorem proving and diagrammatic proofs. In: Proceedings of the European Joint Conferences on Theory and Practice of Software (ETAPS) Satellite Workshop on User Interfaces for Theorem Provers (UITP). Springer, Heidelberg (2005)
24. Wu, W.T.: The characteristic set method and its application. In: Gao, X.S., Wang, D. (eds.) Mathematics Mechanization and Applications, pp. 3–41. Academic Press, San Diego (2000)