TGTP - Thousands of Geometric problems for geometric Theorem Provers

Pedro Quaresma

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

Abstract. *TGTP* (Thousands of Geometric problems for geometric Theorem Provers) is a Web-based library of problems in geometry. The principal motivation in building *TGTP* is to support the testing and evaluation of geometric automated theorem proving (GATP) systems, to help ensure that performance results accurately reflect the capabilities of the GATP system being considered. A common library of problems is necessary for meaningful system evaluations and comparisons, its size is important if the production of statistically significant results is intended. The problems are stored in *TGTP* in an XML-based format for constructive descriptions of geometrical figures and geometrical proofs. It aims, in a similar spirit of *TPTP* and other libraries, is 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.

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) [7] and *SMT-lib* (for satisfiability modulo theory) [2,1] and libraries of problems such as *SATLIB* (for propositional logic) [13], *TPTP* (for predicate logic) [24,25], etc. Such efforts, standards, and libraries are fruitful for easier exchange of problems, ideas, and even program code. However, this is often very demanding and there are not many systems smoothly integrating libraries of problems, theorem provers, and real-world applications.

In the rest of this paper we present *TGTP*[0] (Thousands of Geometric problems for geometric Theorem Provers) which is a Web-based library of GATP test problems. It is a comprehensive common library of problems with a significant

Published as: Pedro Quaresma, Thousands of Geometric problems for geometric Theorem Provers (TGTP), P. Schreck, J. Narboux, and J. Richter-Gebert (Eds.), ADG 2010, LNAI 6877, pp. 169-181. Springer, Heidelberg, 2011. DOI

[0] http://hilbert.mat.uc.pt/TGTP

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.
- 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 170 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. Useful background information, such as an overview of GATP application domains, is provided.
- documents each problem. This contributes to the unambiguous identification of each problem.
- provides a mechanism for adding new problems.
- provides a Workbench for an easy testing of any given conjecture.

There are several systems integrating dynamic geometry softwares (DGS), GATPs, and a set of examples. For example: *Java Geometry Expert*[1] (*JGEX*) is a system that combines dynamic geometry, automated geometry theorem proving, visual dynamic presentation of proofs. It contains a large set of examples of proofs; *GEOTHER* is an environment for manipulating and proving geometric theorems implemented in Maple and contains a collection of theorems in both elementary and differential geometry [12,26]; *Ludi Geometrici*[2] 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*[3] 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 [22].

Many of the DGSs, e.g. *GeoGebra*[4] [9], *Cabri*[5], *Cinderella* [18,23,6], etc, DGSs/GATPs, e.g. *GCLC* [15], *GeoView* [3], *GeoProof* [19], *Geometry Explorer* [27], *MMP/Geometer* [11,10], *GEX* [10], *Discover* [4], and also GATPs like *Theorema* [5] come with a (some times, large) set of examples.

However none of this systems try to provide a common platform for meaningful system evaluations and comparisons.

Paper overview. Section 2 briefly discusses the list of problems, the domain of our system. Section 3 talks about the system and different components: the Web-interface, the the GATPS formats and a common format proposal, the reference

[1] `http://www.cs.wichita.edu/~ye/`
[2] `http://www.polarprof.org/geometriagon/`
[3] `http://hilbert.mat.uc.pt/GeoThms/`
[4] `http://www.geogebra.org/cms/`
[5] `http://www.cabri.com/`

mechanism, the list of problems and the test results. Section 4 discusses further work, and in section 5 some final conclusions are drawn.

2 TGTP

TGTP is a library of problems, in geometry, for GATP systems. *TGTP* aims to supply the automated reasoning in geometry community with a comprehensive library of GATP test problems, in order to provide an overview and a simple, unambiguous reference mechanism. A common library of problems is necessary for meaningful system evaluations and comparisons, its size is important if the production of statistically significant results is intended.

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.

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.

It 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 [21] and in the i2g common file format [8], extending this last format allowing it to cope with conjectures. From this common format for geometric conjectures converters will be written, providing the GATPs code whenever a specific realization was not given.

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

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 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 Figure 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, contributers and the administrator. The administrator has access to a simple interface that allows to see logging information and to do some administrative duties without going directly to the code and/or the database.

The anonymous/regular user has access to the "public" interface. All the access is given in term of "see but do not touch", exception to this is the WORKBENCH where any user can test the problems with the already installed provers, in this section it is also possible to keep a personal scrapbook of problems. The scrapbook it is a per user list of problems, 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 downloads.

The contributers will add to the regular users the ability to add new problems, i.e., in the section "Problems List" the contributers will have, adding to the normal options of anonymous/regular users, the possibility of submit new problems and/or alter the existing ones (see markers (a) and (c) in Figure 1)

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

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

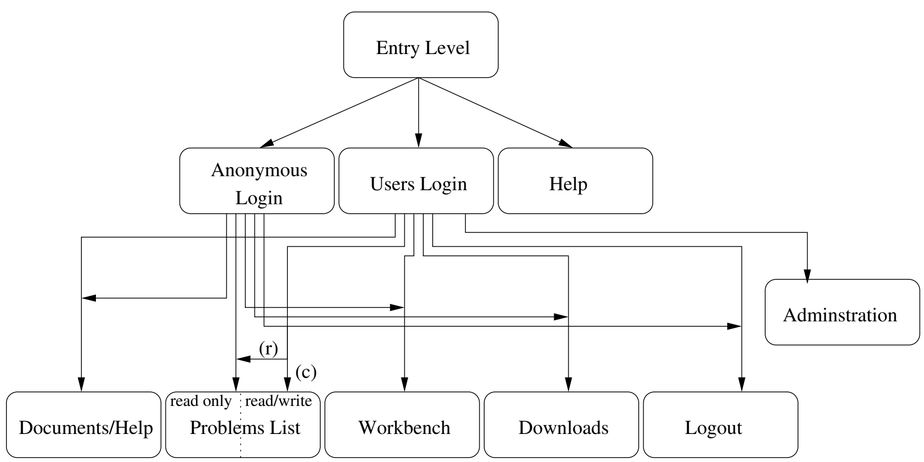

Fig. 1. Structure of the Web-page

Apart from the *Administration* and *Logout* section the interface is divided in four main sections. The *Administration* is the section reserved for administrative duties, the *Logout* section is for a well-behave exit of the interface closing the Web-session and registering some information about the time spent by the user in the system. I will describe the other sections now.

The *Documents/Help* contains documents, for instance, a list of bibliographic references containing all the bibliographic references about GATPs (in BibTeX format) cited, or not, in the problems; a list of provers and a list of authors with informations about GATPs and its authors. This information is introduced by the *TGTP*'s contributers.

This section contains also the performance information regarding the GATPs and the list of problems: number of attempts, number of proofs 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 (only for those cases where the GATP process was not killed by the system after reaching the 600s time limit), the minimum time, the maximum time, and the average time. The information of each individual proof attempt is also displayed (see section 2.5).

Finally, in this section it will be placed all the information regarding the use of *TGTP*: the manuals, frequently asked question list, How-Tos.

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 name of the problem, the name of the problem, a short description (if present), and the number of Proofs succeeded and the number of proofs 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 to easily reading: its identification name, the submission date, its name, a short description and a formal statement (in LaTeX format), and for each proof attempt its status, the GATP used, and the the GATP code.

The contributors will have the possibility to update/alter the info on every existing problem in the database and also a link to a "Add a new Problem" page where they can insert, after a validation step to sieve similar problems already in the database, a new problem. It will be also possible to submit a list of new problems for a bulk insertion into the database, the automatic processing of the list is done with the help of a given XML-format, specified by the author (see Appendix A) to allow this kind of submission. This option it is still a working in progress section.

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

The *Workbench* is a place where 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 want to submit to one of the GATPs that are available in the server, for now GCLCprover [16] and CoqAM [19]. The GATP are called with a 600s time limit and after a successful run, or after 600s, the results of the proof are made available.

Any user has, in this section, access to a personal scrapbook where he/she can save his/her personal conjectures. This scrapbook is unique to every user, exception to the anonymous users that, because they are anonymous, share all the same scrapbook.

In this section is also possible to choose any of the problems in the list of problems. When chosed, from the list of problems or from the scrapbook the problem is loaded in the Web-editor and after that can be tested in the same way the new problems are.

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 [20]. It is also possible to download the Entity-Relationship diagram that describe the database.

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 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 around the problems/conjectures an its proofs (see Figure 2. For each problem, the proofs regarding that problem are kept, for each of them we can access the information about the GATP used and also the measures of efficiency. At a given time a snapshot of the results are taken and a date tag attached, this results are kept in the "Performance Information" section and are used for evaluation of the improvements in the GATP state of the art along the time.

The `CodeTmpProver` table is used to support the user's scrapbook. The `TGTP` table is used to save the information of the system's version to historical references.

Given the fact that *TGTP* shares with the *GeoThms* system the database of problems we can have, for many of the conjectures but not necessarily for all, the DGS construction also, but this is only available in the *GeoThms* system.

2.4 Queries

The list of problems can be queried in two ways: a simple query using the MySQL regular expressions [20], this query is done over the `name` attribute of the table `Conjectures`, the user will provide a word to be searched and this is matched against any of the words in the list of words that constitute the conjectures names.

Another, more powerful, query is done using the full-text search of MySQL [20] this is done over the attributes `name, description, shortDescription` of the table `Conjectures` and allows, for a given input sentence, to get the list of most similar sentences in either the `name`, the `description` or the `shortDescription` attributes of the different problems.

2.5 Performance Information

The *TGTP* database contains now (2011/02/03) 175 problems and contains results of proof attempts from two GATP: CoqAM [19], and GCLCprover [16],

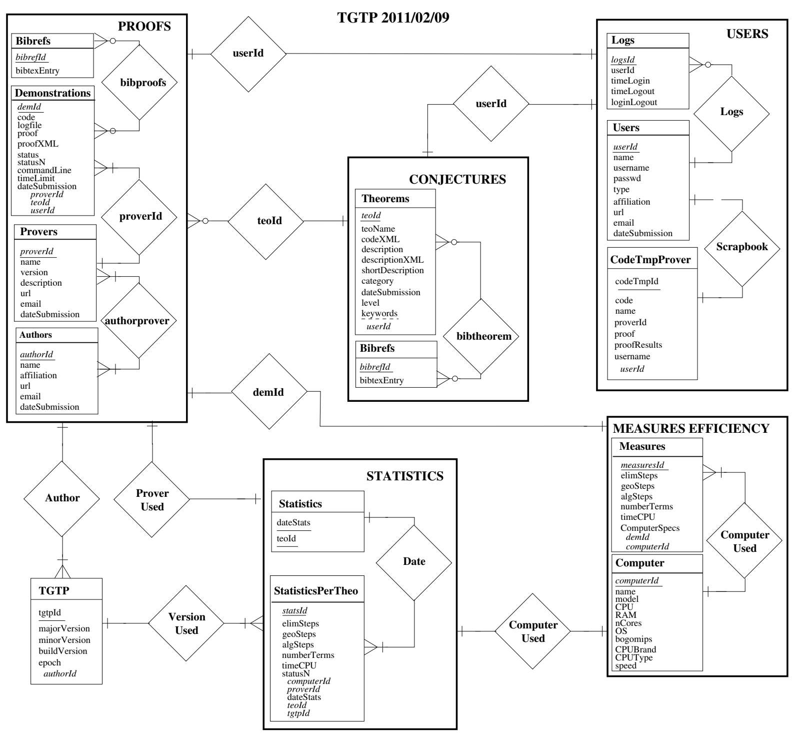

Fig. 2. Structure of the Data Base (E-R Diagram)

covering the methods: Wu's Method [28], Gröbner Basis Method [17] and Area Method [14].

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, whenever one of this situations occurs a new set of performance values is taken, that is, the computer it is run on all the problems versus all the GATPs (whenever the code for that given problem/GATP is present). This it will be improved to make a run only for the new situations whatever they are, keeping the old ones without change, the exception to this in the change of the computer that serves the *TGTP* system.

The new problems adds to the existing ones so it will be possible 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 problem the script will verify the existence in the data base of GATP's code for that problem and it will run the GATP on it. All the proofs attempts have a time limit of 600s after which the process is killed by the operating system. The proof status, and its code, could be: "Time-out: Failed to prove the conjecture" (4), when the process is killed before it reach and end; "The conjecture out of scope of the prover" (6) 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 [14] for details); "Maximal number of proof steps reached: Failed to prove the conjecture" (5) a limit that some GATPs (for example GCLC AM), have themselves; "Failed to prove the conjecture" (3); "Disproved" (2) and "proved" (1).

	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

Table 1. Results of Proof Attempts

Apart from this, per problem, results some overall values are also collected (see Table 2). For each GATP the number of proofs 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, taken by the script only for the cases where the GATP ends in a normal way, e.g. the process is not killed by the script, the minimum time needed, the maximum time needed, and the average time.

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 take 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 results needed.

	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

Table 2. Overall Results

3 Common File Format for Conjectures

In [21] it is described an XML-suite for constructive descriptions of geometrical figures and geometrical proofs, this is used in the *GeoThms* system to provide a common format for its list of problems. In *GeoThms* the conversion of this format to the DGSs/GATPs format is done via XSLT files. A specific DTD document defines syntactical restrictions for constructing descriptions, this DTD document can then be used, in conjunction with the generic XML validation mechanism, for verifying whether a given description of a geometrical construction is legal [21].

Since then the i2g common file format of the Intergeo consortium was specified, the i2g is a file format designed to describe any construction with a DGS [8].

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 [8] the Content Dictionaries mechanism of OpenMath 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 common format to all the GATPs formats available in the *TGTP* system.

```
definitions,proof,status,NDGconditions,prover_report
proof_step | lemma
equality,explanation,semantics
proof,status
```

4 State of the art

There are several systems integrating dynamic geometry softwares (DGS), GATPs, and a set of examples. For example:

Java Geometry Expert[6] (*JGEX*) is a new, Java version of *GEX*. *JGEX* is being developed from 2004, by Shang-Ching Chou, Xiao-Shan Gao, and Zheng Ye. *JGEX* combines dynamic geometry, automated geometry theorem proving, and, as its most distinctive part, visual dynamic presentation of proofs. *JGEX* implements the following methods for geometry theorem proving: Wu's method, the Gröbner basis method, the full-angle method, the deductive database method, the area method and the vector method are still under development. It contains a large set of examples of proofs.

GEOTHER is an environment for manipulating and proving geometric theorems implemented in Maple, with drawing routines and the interface in Java. *GEOTHER* can work with a menu-driven graphic user interface and contains a collection of theorems in both elementary and differential geometry [12,26].

Ludi Geometrici[7] has a vast library of problems in the area of classical constructive (ruler and compass only) Euclidean geometry. It does not provide an GATP. A user can only perform valid steps in the construction, using only a limited set of tools, and in this way the system is capable to recognise whenever a user has reach a solution of a problem. No formal proofs are provided.

GeoThms[8] is a Web workbench in the field of constructive problems in Euclidean geometry. It links dynamic geometry software, geometry automatic theorem provers, and a library of geometry problems (geoDB), providing a common Web interface for all these tools [22]

Many of the DGSs (e.g. *GeoGebra*[9] [9], *Cabri*[10], *Cinderella* [18,23,6], etc.), DGSs/GATPs (e.g. *GCLC* [15], *GeoView* [3], *GeoProof* [19], *Geometry Explorer* [27], *MMP/Geometer* [11,10], *GEX* [10], *Discover* [4]), and also GATPs like *Theorema* [5] come with a (some times, large) set of examples. However none of them try to provide a common platform for meaningful system evaluations and comparisons.

[6] http://www.cs.wichita.edu/~ye/
[7] http://www.polarprof.org/geometriagon/
[8] http://hilbert.mat.uc.pt/GeoThms/
[9] http://www.geogebra.org/cms/
[10] http://www.cabri.com/

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 problem collection, 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.

A List of Problems XML Format

The list 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 can also be used by the *TGTP* users to get the information from the system.

The XML format has the necessary tags to describe (an load it into the database) any given problem. The tags are self-explanatory, the example below describe the format. The author of this text is open to any suggestion/improvement to this format that the readers should be willing to provide.

```
<results>
 <result>
  <userid>
   Contributer Id (mandatory)
  </userid>
  <theoname>
   Theorem Name (mandatory)
  </theoname>
  <description>
   Theorem statement in LaTeX format (optional)
  </description>
  <shortDescription>
   Theorem statement in text format (optional, but desirable)
  </shortDescription>
  <observations>
   Observations (optional)
  </observations>
  <figcode>
   DGS code for the rendering of the Geometric Construction (optional)
  </figcode>
  <proofscode>
   GATPs code
```

```
  </proofscode>
  <bibtexentry>
   Bibliographic entry, in BibTeX format (optional)
  </bibtexentry>
 </result>
 ...
</results>
```

B The Common Format for GATPs

Statements for the basic sorts of conjectures are given in the following table:

```
points A and B are identical:   identical A B
points A, B, C are collinear:   collinear A B C
AB is perpendicular to CD:      perpendicular A B C D
AB is parallel to CD:           parallel A B C D
O is the midpoint of AB:        midpoint O A B
AB has the same length as CD:   same_length A B C D
points A, B, C, D are harmonic: harmonic A B C D
```

All these sorts of conjectures can also be expressed in terms of geometry quantities. Geometry quantities provide more general way for stating conjectures.

In GCLC, geometry quantities are written as in the following examples:

```
ratio of directed segments      sratio P Q A B
signed area (arity 3)           signed_area3 A B C
signed area (arity 4)           signed_area4 A B C D
Pythagoras difference (arity 3)  pythagoras_difference3 A B C
Pythagoras difference (arity 4)  pythagoras_difference4 A B C D
```

A conjecture to be proved is given as argument to the prove command. It has to be some of the basic sorts of conjectures (see Section 6.2), or it has to be of the form L = R, where L and R are expressions over geometry quantities. The conjecture can involve geometry quantities (only) over points already introduced (by a subset of commands) within the current construction. Geometry quantities can be combined together into more complex terms by operators for addition, multiplication and division. Operators are written in textual form as in the following table:

=	equality
+	sum
·	mult
/	ratio

The conjecture and all its subterms are written in prefix form, with brackets if needed. For instance,

SA1 B1 A = S A1 B1 B

is given to be proved in the following way:

```
prove  equal  signed_area3 A_1 B_1 A
                   signed_area3 A_1 B_1 B
```

```
prove  equal  mult  mult  sratio A F F B
                             sratio B D D C
                      sratio C E E A
              1
```

```
prove  equal  signed_area3 A_1 B_1 A
               signed_area3 A_1 B_1 B
```

```
conjecture
prove
equality
expression
signed_area3.
number
...
signed_area4,
pythagoras_difference3
pythagoras_difference4

lemma (proof,status)

expression (number| sum| mult| fraction| segment_ratio|
signed_area3 | signed_area4 | pythagoras_difference3 |
pythagoras_difference4)

equality (expression,expression)

NDGconditions

prover_report (elimination_steps,geometrical_steps,
algebraic_steps,total_number_of_steps,time)

<conjecture>
```

```
 <prove proof_level="1" proof_limit="10000">
  <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>
```

References

1. Clark Barrett, Aaron Stump, and Cesare Tinelli. The Satisfiability Modulo Theories Library (SMT-LIB). `www.SMT-LIB.org`, 2010.
2. Clark Barrett, Aaron Stump, and Cesare Tinelli. The SMT-LIB Standard: Version 2.0. In A. Gupta and D. Kroening, editors, *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories (Edinburgh, England)*, 2010.
3. Yves Bertot, Frédérique Guilhot, and Loïc Pottier. Visualizing geometrical statements with GeoView. *Electronic Notes in Theoretical Computer Science*, 103:49–65, 2004.
4. Francisco Botana and José L. Valcarce. A dynamic-symbolic interface for geometric theorem discovery. *Computers and Education*, 38:21–35, 2002.
5. B. Buchberger, A. Craciun, T. Jebelean, L. Kovacs, T. Kutsia, K. Nakagawa, F. Piroi, N. Popov, J. Robu, M. Rosenkranz, and W. Windsteiger. Theorema: Towards computer-aided mathematical theory exploration. *Journal of Applied Logic*, 4(4):470–504, 2006.
6. Cinderella Web site. `http://www.cinderella.de`.
7. DIMACS. Satisfiability suggested format. `ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/doc/`, May 1993.
8. Santiago Egido, Maxim Hendriks, Yves Kreis, Ulrich Kortenkamp, and Daniel Marquès. i2g Common File Format Final Version. Technical Report D3.10, The Intergeo Consortium, 2010.
9. Karl Fuchs and Markus Hohenwarter. 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*, pages 128–133, Pécs, Hungary, 2004.
10. Xiao-Shan Gao. Software packages. `http://www.mmrc.iss.ac.cn/~xgao/software.html`.
11. Xiao-Shan Gao and Qiang Lin. MMP/Geometer - a software package for automated geometric reasoning. *Lecture Notes in Computer Science*, 2930:44–66, 2004.
12. GEOTHER web site. `http://www-calfor.lip6.fr/~wang/GEOTHER/`.
13. Holger Hoos and Thomas Stützle. SATLIB: An online resource for research on SAT. In T.Walsh I.P.Gent, H.v.Maaren, editor, *SAT 2000*, pages 283–292. IOS Press, 2000.

14. P. Janičić, J. Narboux, and P. Quaresma. The Area Method: a recapitulation. *Journal of Automated Reasoning*, (to appear), 2011. DOI: 10.1007/s10817-010-9209-7.
15. Predrag Janičić. GCLC — a tool for constructive euclidean geometry and more than that. *Lecture Notes in Computer Science*, 4151:58–73, 2006.
16. Predrag Janičić and Pedro Quaresma. System Description: GCLCprover + GeoThms. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, volume 4130 of *LNAI*, pages 145–150. Springer, 2006.
17. D. Kapur. Using Gröbner bases to reason about geometry problems. *Journal of Symbolic Computation*, 2(4):399–408, 1986.
18. Ulrich Kortenkamp and Jürgen Richter-Gebert. Using automatic theorem proving to improve the usability of geometry software. In *Procedings of the Mathematical User-Interfaces Workshop 2004*, 2004.
19. Julien Narboux. A graphical user interface for formal proofs in geometry. *Journal of Automated Reasoning*, 39:161–180, 2007.
20. Oracle. *MySQL 5.5 Reference Manual*, 5.5 edition, January 2011. revision: 24956.
21. P. Quaresma, Tomašević J. Janičić, P., M. V.-Janičić, and D. Tošić. *Communicating Mathematics in The Digital Era*, chapter XML-Bases Format for Descriptions of Geometric Constructions and Proofs, pages 183–197. A. K. Peters, Ltd., 2008.
22. Pedro Quaresma and Predrag Janičić. Geothms – a Web System for Euclidean Constructive Geometry. *Electronic Notes in Theoretical Computer Science*, 174(2):35 – 48, 2007.
23. Jürgen Richter-Gebert and Ulrich Kortenkamp. *The Interactive Geometry Software Cinderella*. Springer, 1999.
24. Geoff Sutcliffe. The TPTP problem library. `http://www.cs.miami.edu/~tptp`.
25. Geoff Sutcliffe. The TPTP problem library and associated infrastructure. *Jounal of Automated Reasoning*, 43(4):337–362, 2009.
26. Dongming Wang. GEOTHER 1.1: Handling and proving geometric theorems automatically. *Lecture Notes in Artificial Intelligence*, 2930:194–215, 2004.
27. Sean Wilson and Jacques Fleuriot. 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, 2005.
28. W.-T. Wu. The characteristic set method and its application. In X.-S. Gao and D. Wang, editors, *Mathematics Mechanization and Applications*, pages 3–41, San Diego, CA, 2000. Academic Press.