TR ongoing work ISSN 0874-338X

The i2GATPformat

(Revision:45; Last changed on:2012-02-27 22:27:57 +0000 (Seg, 27 Fev 2012))

Pedro Quaresma
Department of Mathematics
University of Coimbra, Portugal

Center for Informatics and Systems of the University of Coimbra

The I2GATPformat

(Revision:45; Last changed on:2012-02-27 22:27:57 +0000 (Seg, 27 Fev 2012))

Pedro Quaresma
CISUC/Department of Mathematics
University of Coimbra
3001-454 Coimbra, Portugal
E-mail: pedro@mat.uc.pt

02-2012

Abstract

A common setting for interoperable interactive geometry was already proposed, the i2g format, but it lacks to this format the conjectures and proofs counterpart. A common format capable of linking all the tools in the field of geometry is missing.

In this document an extension to the i2g format is proposed. An extension capable of describe not only the geometric constructions but also geometric conjectures.

This is a support document to the article "A Format for Conjectures in Geometry" submitted to MKM2012.

Chapter 1

Introduction

This technical report serves as support for the document "A Format for Conjectures in Geometry",submitted to MKM 2012, providing a more detailed account of the symbol list for the different XML files in the I2GATP format.

Paper overview. In Section 2 the overall structure of the new format is described. In Sections 3 and 4 implementations issues are discussed and one example is presented. Finally in Section 5 some final conclusions are drawn and future work is discussed.

Chapter 2

Overall Architecture

2.1 Structure of i2gatp format

The structure of the i2gatp format has a root node, the *problem* node and four sub-nodes: *information*, *construction*, *conjecture* and *proofs* (see Figure 2.1).

Each one of these sub-nodes can be empty (it is possible to have all nodes empty at the same but that is completely useless).

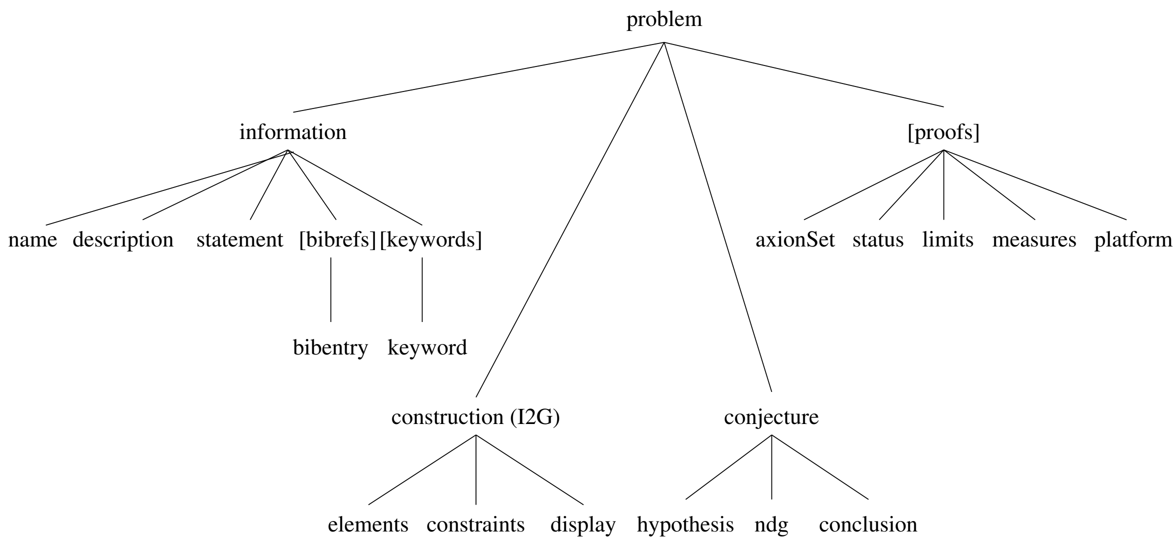


Figure 2.1: Structure of the i2gatp File Format

2.1.1 Information

The *information* sub-tree contains all the generic (human) information about the problem. The *name* of the problem; a brief, informal, *description* of the problem; a rigorous mathematical description (*statement*) of the problem; a list of bibliographic references; a list of keywords.

2.1.2 Construction

The *construction* sub-tree contains the construction in the I2G format, i.e. the file `intergeo.xml`. The I2G format specifies three sub-nodes: *elements* for the free objects; *constraints* for the objects fixed by construction constraints and *display* for the display details.

2.1.3 Conjecture

This is the core of the I2GATP format. In here the *hypothesis*, the *ndg* (non-degenerate conditions) and the *conclusion* establishing the conjecture to be proved are specified. The non-degenerate conditions could be a side-effect of the proving process, e.g. automatically generated by a GATP based in the area method, or provided manually.

2.1.4 Proofs

For a given problem/conjecture we can have many proof attempts: different approaches, for instance synthetic proof versus algebraic proof; different methods, Gröbner bases method versus Wu's method; different GATPs, GCLCprover versus CoqAM, and all the possible combinations of this three different aspects. In conclusion this is a node of type list.

Each proof attempt will be named accordingly to the GATP used, its version and the method used, e.g. `proofGCLC9.0AreaMethod`.

Each individual proof node will have: the information regarding the axiom set and rules of inference, e.g. the *area method*; the status of the proof, e.g. *proved*; the computational constraint regarding the proof attempt made by the GATP, e.g. maximum CPU time and RAM space allowed by the system; the proof metrics, e.g. number of proof steps (area method) and the platform used when doing the proof, e.g. CPU, RAM, and other details about the computational platform.

Given the fact that the proofs produced by different GATPs/Methods are, and should continue to be, quite different we do not try to create a common formats for the proofs. The outcomes produced by the different GATPs will be kept as they are produced (see the *container* in section 2.2).

2.2 The container

Following the ideas of the I2G common format all the file related to the I2GATP format will be packed in a single compressed file, the *container*, which is nothing more then a I2G container with three additional directories (`information`, `conjecture` and `proofs`). This means that it will be possible to extract the I2G container out of this file, it will be a simple question of unpack the file, erase the additional directories and repack, if needed, the resulting files.

The structure of the container follows closely the structure of the I2GATP format. The `information`, `construction` and `conjecture` directories will contain the files `information.xml`, `intergeo.xml` and `conjecture.xml` respectively. The directory `construction` may also contain the rendering of the construction in various graphical formats (e.g. PDF, SVG, PNG, etc.).

The directory `proofs` will contain as many sub-directories, as proofs attempts were made for the problem in question. The naming convention follows the ideas in the I2G format, that is, after the prefix "proof", the name of the GATP, its version, and finally the method used.

information/	mandatory
information/information.xml	optional
construction/	mandatory
construction/intergeo.xml	optional
construction/preview.pdf	optional
construction/preview.svg	optional
construction/(. . .)	
conjecture/	mandatory
conjecture/conjecture.xml	optional
proofs/	mandatory
proofs/proof<GATP><Version><AxiomSet>/	optional
proofs/proof<GATP><Version><AxiomSet>/proofInfo.xml	optional
proofs/proof<GATP><Version><AxiomSet>/proofOutput.pdf	optional
proofs/proof<GATP><Version><AxiomSet>/(. . .)	
metadata/	optional
metadata/i2g-lom.xml	optional
resources/	optional
resources/<image_files>	optional
resources/(. . .)	
private/	optional
private/<domain-name>	optional
private/<domain-name>/<files>	optional

Table 2.2: The I2GATP container

In each of this sub-directories the file `proofInfo.xml` will contain the information regarding the proof attempt. This directory may also contain files with the rendering of the proof in different formats (e.g. PDF, HTML, etc.).

The remaining directories follow the structure of the I2G format and can be used to place additional contents produced by the GATPs.

Following the I2G conventions, the suggest naming convention to the container is problem<problem_name>

In the next section the symbol lists, i.e. the tags proposed to this XML-format, is described.

2.3 Symbol Lists

2.3.1 information.xml

Generic information about the problem. All fields, except the *name*, may be empty.

Tag Name	Description	Type
name	name of the problem	text
description	brief, informal, description of the problem	text
statement	rigorous mathematical description of the problem.	text, MathML format
bibrefs	list of bibliographic citations	list
bibentry	bibliographic reference	text, BibTeXml format
keywords	list of keywords	list
keyword	keyword	text

2.3.2 conjecture.xml

Contains all the info regarding the conjecture

Tag Name	Description	Type
hypothesis	pre-conditions to the conjecture	text
ndg	non-degenerated conditions	
conclusion	statement to be proved	text

The hypothesis and the statement would contain inside it many other symbols. The following list of symbols is not exhaustive, some of them are already contained in some other CD (not yet checked).

Tag Name	Description	Type
defines	I2G CD	
constructions	I2G CD	
hypothesis	I2G CD + (maybe) other symbols	
not_equal	elements of hypothesis	

not_parallel	elements of hypothesis	
equal	arith CD	
plus	arith CD	
minus	arith CD	
mult	arith CD	
div	arith CD	
collinear	I2G CD	
perpendicular	I2G CD	
parallel	I2G CD	
midpoint	I2G CD	
same_length	I2G CD	
harmonic	I2G CD	
segment_ratio	area method	
signed_area_3	area method	
signed_area_4	area method	
pythagoras_diff_3	area method	
pythagoras_diff_4	area method	

2.3.3 proofInfo.xml

Contains all the information regarding a proof attempt for given problem. This is a record of the conditions under which the proof was attempted.

The directory **proofs** can be empty. If that happens it means that the container has only the geometric construction, apart the information node.

Tag Name	Description	Type
proof_info	a given proof	text

Two attributes: the `GATP`, the prover used and `Version` its version.

axiomSet node

The theory used in the proof, e.g. the area method [2].

Tag Name	Description	Type
axiomSet	The theory used in the proof	text (Area Method | Wu's Method | Gröbner Basis Method | $\cdots$)

The different outputs of a proof attempt. This section could be empty, e.g. a problem without any proof attempt yet.

limits the different limits that could be imposed to the running of the GATP.

It may be empty.

measures measures of efficiency in terms of the hardware but also the method used.

It may be empty.

platform the hardware/operating system used in the proof.

It may be empty.

The following list of symbols is not exhaustive.

Tag Name	**Description**	**Type**
result	proof status	proved | disproved | not proved | out of scope | failed to prove
limits		tag
time_limit_seconds	maximun CPU time (in seconds) limit imposed to the GATP.	integer
space_imit_MiB	maximum RAM (in MiB) space limit imposed to the GATP	float
iterations_limit	limit in the number of iterations imposed to the GATP	integer
measures		tag
CPU_time	CPU time, in seconds, used by the GATP	float
RAM_space	RAM spaced used, in MiB, used by the GATP	float
elimination_steps	number of elimination steps (area method), optional	integer
geometrics_steps	number of geometric steps (area method), optional	integer
algebraic_steps	number of algebraic steps (area method), optional	integer
number_terms_largest_polynomial	number of terms in the largest polynomial (Wu's method and Grbner Basis method), optional	integer
computer_name	name (or IP) of the computer used in the proof, optional	text

model	model of the computer, optional	text
CPU_type	type of the CPU, optional	text
clock_speed	clock speed, in GHz, optional	float
number_cores	number of cores (cores and hyper-threading), optional	integer
RAM	size of the computer's RAM, in MiB, optional	integer
speed	measures of velocity, in bogoMIPS, optional	float
operating_system	Operating system, optional	text

The meanings for the above mentioned values are:

proved the statement was proved to be true.

disproved the statement was proved to be false.

not proved the GATP didn't reach a conclusion.

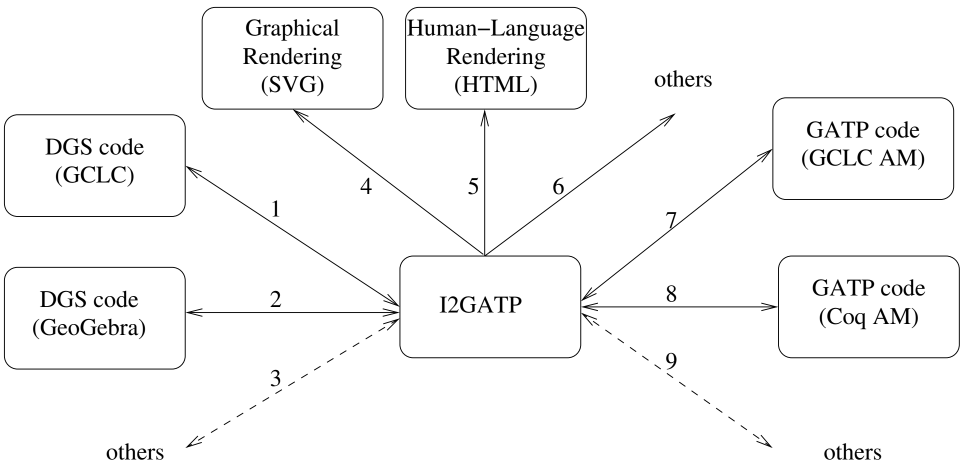
out of scope the statement is out of the scope of the GATP.

failed to prove due to a time, space, or other type of constraint, the proof was not completed.

Chapter 3

Implementation

Having defined a XML format for geometric constructions and conjectures its usefulness depends on its support from other tools, i.e. the capability of tools such as DGSs (see [1] to the list of tools already supporting the I2G format) and GATPs to export to the I2GATP format and, of course, its support to other tools in the shape of converters from I2GATP format to the internal format of tools such as the DGSs and GATPs (see Figure 3.1).



1 – From/to GCLC to/from I2G(ATP)
2 – From/to GeoGebra to/from I2G(ATP)
3 – From/to DGS to/from I2G(ATP)
4 – SVG rendering
5 – HTML rendering
6 – other: proofs; bibrefs., etc.
7 – From/to I2GATP to/from GCLC AM
8 – From/to I2GATP to/from Coq AM
9 – From/to I2GATP to/from GATP

Figure 3.1: Conversions From/To I2GATP To/From Geometric Tools

Using the *TGTP* project as a catalyst for this task I will try to provide (working in conjunction with the authors of the tools).

- converters from dynamic DGSs and GATPs tools (GCL language, Coq AM, etc.) to I2GATP format; these converters will take the form of C++ and/or Java libraries written with the help of the Xerces library.
- converters from I2GATP format to DGSs and GATPs tools (GCL language, Coq AM, etc.) these converters will be written as XSLT files) or using the Xerces library.

- a converter from the i2g format to the i2gatp format. This is a simple script capable of converting the container from one format to the other. From i2g to i2gatp a simple inclusion (creating empty directories where needed), and in the other direction a simple suppression of the extra directories and files.
- The renderings of the construction (e.g. the SVG rendering), this will allow to free the tools of this task providing, in this way, a common rendering mechanism. The rendering of the problem/conjecture in, e.g. natural language format [3].

The *TGTP* and *GeoThms* servers will use the i2gatp as its base format, providing converters to and from the different GATPs.

3.1 Producers

Programs that export to i2gatp

- GCLCprover → i2gatp
- CoqAm → i2gatp

Plus: DGSs + GATPs + CASs

3.2 Converters

Build a set of Converters, xslt mechanism? flex mechanism?
i2gatp ⟷ CoqAM
i2gatp ⟷ GCLCproverAM
i2gatp ⟷ GCLCproverWuM
i2gatp ⟷ GCLCproverGBM
...

3.3 Readers

TGTP?

Chapter 4

An Example

Using the Ceva's Theorem as an example, the contents of the I2GATP container will be described. For the sake of brevity, in the XML files some parts were substituted by ellipses.

Theorem 1 (Ceva‘s Theorem) *Let ΔABC be a triangle and P be any point in the plane. Let $D = AP \cap CB$, $E = BP \cap AC$, and $F = CP \cap AB$. Show that:*

$$\frac{\overline{AF}}{\overline{FB}}\frac{\overline{BD}}{\overline{DC}}\frac{\overline{CE}}{\overline{EA}} = 1$$

P should not be in the lines parallels to AC, AB and BC and passing through B, C and A respectively.

The container will be a zip file with name `problemGEO0001.zip` (*TGTP* classification). Its contents will be:

Information Directory this directory will contain the file `information.xml` only.

```
<information>
 <name>
  Ceva's Theorem
 </name>
 <description>
  Given a triangle ABC, and points D, E, and F that lie on lines
  BC, CA, and AB respectively, the theorem states that lines AD, BE
  and CF are concurrent if and only if AF/FB*BD/DC*CE/EA=1.
 </description>
 <statement>
  <!--l. 9-->
  <p class="noindent" >Let
  <!--l. 9-->
  <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
  <mi>&#x0394;</mi><mi>A</mi><mi>B</mi><mi>C</mi></math> be a triangle
  and
  <math xmlns="http://www.w3.org/1998/Math/MathML"  display="inline" >
  <mi>P</mi></math>
  be any point in the plane. Let
  <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
  <mi>D</mi> <mo class="MathClass-rel">=</mo>
  <mi>A</mi><mi>P</mi> <mo class="MathClass-bin">&#x2229;</mo>
  <mi>C</mi><mi>B</mi></math>,
  <!--l. 10-->
  <math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
  <mi>E</mi> <mo class="MathClass-rel">=</mo> <mi>B</mi><mi>P</mi>
```

```
<mo class="MathClass-bin">&#x2229;</mo><mi>A</mi><mi>C</mi></math>, and
<!--l. 10-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>F</mi> <mo class="MathClass-rel">=</mo> <mi>C</mi><mi>P</mi>
<mo class="MathClass-bin">&#x2229;</mo> <mi>A</mi><mi>B</mi></math>.
Show that: <!--tex4ht:inline--></p>
<!--l. 11-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="block" >
<mfrac><mrow><mover accent="false" class="mml-overline"><mrow>
<mi>A</mi><mi>F</mi></mrow><mo accent="true">&#x00AF;</mo></mover></mrow>
<mrow><mover accent="false" class="mml-overline"><mrow><mi>F</mi>
<mi>B</mi></mrow><mo accent="true">&#x00AF;</mo></mover></mrow></mfrac>
<mo class="MathClass-bin">&#x00D7;</mo><mfrac><mrow>
<mover accent="false" class="mml-overline"><mrow><mi>B</mi><mi>D</mi></mrow>
<mo accent="true">&#x00AF;</mo></mover></mrow>
<mrow><mover accent="false" class="mml-overline"><mrow><mi>D</mi>
<mi>C</mi></mrow><mo accent="true">&#x00AF;</mo></mover></mrow></mfrac>
<mo class="MathClass-bin">&#x00D7;</mo><mfrac><mrow>
<mover accent="false" class="mml-overline"><mrow><mi>C</mi><mi>E</mi></mrow>
<mo accent="true">&#x00AF;</mo></mover></mrow>
<mrow><mover accent="false" class="mml-overline"><mrow><mi>E</mi><mi>A</mi>
</mrow><mo accent="true">&#x00AF;</mo></mover></mrow></mfrac>
<mo class="MathClass-rel">=</mo> <mn>1</mn>
</math>
<!--l. 15-->
<p class="nopar" >
<!--l. 16-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>P</mi></math> should not be in the lines parallels to
<!--l. 16-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>A</mi><mi>C</mi></math>,
<!--l. 16-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>A</mi><mi>B</mi></math> and
<!--l. 16-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>B</mi><mi>C</mi></math> and passing through
<!--l. 17-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>B</mi></math>,
<!--l. 17-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>C</mi></math> and
<!--l. 17-->
<math xmlns="http://www.w3.org/1998/Math/MathML" display="inline" >
<mi>A</mi></math>
 respectively. </p>
</statement>
<bibrefs>
  <bibtex:entry id="Chou87">
  <bibtex:book>
  <bibtex:title>Mechanical Geometry Theorem Proving</bibtex:title>
  <bibtex:publisher>D. Reidel Publishing Company</bibtex:publisher>
  <bibtex:year>1987</bibtex:year>
  <bibtex:author>Chou, Shang-Ching</bibtex:author>
  <bibtex:address>Dordrecht</bibtex:address>
  </bibtex:book>
 </bibtex:entry>
</bibrefs>
<keywords
 <keyword>parallels</keyword>
 <keyword>triangle</keyword>
</keywords>
</information>
```

Construction Directory this directory is a direct replica of the directory with the same name in the I2G format.

It will contain the `intergeo.xml` file and any file containing a graphical representation of the construction.

The `intergeo.xml` file for this example is given below.

```
<construction>
 <elements>
  <point id="A">
   <euclidean_coordinates>
    <double>60</double>
    <double>10</double>
   </euclidean_coordinates>
  </point>
  <point id="B">
   <euclidean_coordinates>
    <double>30</double>
    <double>90</double>
   </euclidean_coordinates>
  </point>
  <point id="C">
   <euclidean_coordinates>
    <double>80</double>
    <double>90</double>
   </euclidean_coordinates>
  </point>
  <point id="P">
   <euclidean_coordinates>
    <double>55</double>
    <double>75</double>
   </euclidean_coordinates>
  </point>
 </elements>
 <constraints>
  <line_through_two_points>
   <line out="true">bc</line>
   <point>B</point>
   <point>C/point>
  </line_through_two_points>
  <line_through_two_points>
   <line out="true">ab</line>
   <point>A</point>
   <point>B</point>
  </line_through_two_points>
  <line_through_two_points>
   <line out="true">ac</line>
   <point>A</point>
   <point>C</point>
  </line_through_two_points>
  <line_through_two_points>
   <line out="true">ap</line>
   <point>A/point>
   <point>Ppoint>
  </line_through_two_points>
  <line_through_two_points>
   <line out="true">bp/line>
   <point>B</point>
   <point>P/point>
  </line_through_two_points>
  <line_through_two_points>
   <line out="true">cp/line>
   <point>C</point>
   <point>P/point>
  </line_through_two_points>
  <point_intersection_of_two_lines>
```

```
    <point out="true">D<line>
    <line>bc</line>
    <line>ap</line>
   </point_intersection_of_two_lines>
   <point_intersection_of_two_lines>
    <point out="true">E<line>
    <line>ac</line>
    <line>bp</line>
   </point_intersection_of_two_lines>
   <point_intersection_of_two_lines>
    <point out="true">F<line>
    <line>ab</line>
    <line>cp</line>
   </point_intersection_of_two_lines>
  </constraints>
  <display>
   <style ref="A">
    <label>A</label>
   </style>
   <style ref="B">
    <label>B</label>
   </style>
    (...)
  </display>
 </construction>
```

Conjecture Directory The file `conjecture.xml` will contain the hypothesis, the non-degeneracy conditions and the conclusion.

```
<conjecture>
 <hypothesis>
  <parallel>
   <line_through_two_points>
    <line out="true">af</line>
    <point>A</point><point>F</point>
   </line_through_two_points>
   <line_through_two_points>
   <line out="true">fb</line>
   <point>F</point><point>B</point>
   </line_through_two_points>
    (...)
  </parallel>
 </hypothesis>
 <ndg>
  <not_equal>
   <point>F</point><point>B</point>
  </not_equal>
   (...)
 </ndg>
 <conclusion>
  <equality>
   <expression>
    <mult>
     <expression>
      <mult>
       <expression>
        <segment_ratio>
         &lt;segment&gt;
          <point>A</point>
          <point>F</point>
         &lt;/segment&gt;&lt;segment&gt;
          <point>F</point>
          <point>B</point>
         &lt;/segment&gt;
        </segment_ratio>
```

```
        </expression>
        <expression>
          <segment_ratio>
            &lt;segment&gt;
              <point>B</point>
              <point>D</point>
            &lt;/segment&gt;
            &lt;segment&gt;
              <point>D</point>
              <point>C</point>
            &lt;/segment&gt;
          </segment_ratio>
        </expression>
      </mult>
    </expression>
    <expression>
      <segment_ratio>
        &lt;segment&gt;
          <point>C</point>
          <point>E</point>
        &lt;/segment&gt;
        &lt;segment&gt;
          <point>E</point>
          <point>A</point>
        &lt;/segment&gt;
      </segment_ratio>
    </expression>
    </mult>
   </expression>
   <expression>
    <number>1.000000</number>
   </expression>
  </equality>
 </conclusion>
</conjecture>
```

Proofs Directory This directory will contain as many sub-directories as proofs attempts. Each of this directories will have a name referring to the GATP used its version and the method used.

For a proof attempt developed using the GCLCprover, version 9.0, using the area method, the `proofInfo.xml` file will be like this (again, with some parts substituted by ellipses):

```
<proofInfo GATP='GCLCprover' Version='9.0'>
 <axiomSet>
  Area Method
 </axiomSet>
 <outputs>
  <status>
   proved
  </status>
  <limits>
   <maxSecondsCPU>60</maxSecondsCPU>
  </limits>
  <measures>
   <CPUtime>0</CPUtime>
   <eliminationSteps>3</eliminationSteps>
   <geometricsSteps>6</geometricsSteps>
   <algebraicSteps>23</algebraicSteps>
  </measures>
  <platform>
   <computerName>hilbert.mat.uc.pt</computerName>
   <model>x86_64 unknown</model>
   <CPUtype>Intel(R) Pentium(R) 4 CPU 3.00GH</CPUtype>
   <clockSpeed>3.00</clockSpeed>
```

```
    <numberCores>2</numberCores>
    <RAM>2</RAM>
    <bogoMips>5989.8</bogoMips>
    <operatingSystem>2.6.22-3-686-bigmem GNU/Linux</operatingSystem>
   </platform>
  </outputs>
</proofInfo>
```

Chapter 5

Conclusions and Further Work

Questions and future work to be addressed:

- the XML format must be complemented with an extensive set of converters allowing the exchange of information between as many geometric tools as possible;
- the databases queries, as in *TGTP*, raise the question of selecting appropriate keywords. A fine grain index and/or an appropriate geometry ontology should be addressed;
- the I2GATP format do not address the proof itself. Should we try to create such a format? As far as I see the GATPs produce proofs in quite different formats, maybe the construction of such unifying format it is not possible and/or desirable in this area.
- to extend the database of geometric constructions within *GeoThms* and *TGTP*.

Bibliography

[1] The Intergeo Consortium. Intergeo implementation table. `http://i2geo.net/xwiki/bin/view/I2GFormat/ImplementationsTable`.

[2] P. Janičić, J. Narboux, and P. Quaresma. The Area Method: a recapitulation. *Journal of Automated Reasoning*, 2011. online.

[3] 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.