XML-based Format for Geometry

XML-based Format for Descriptions of Geometrical Constructions

and Geometrical Proofs

Pedro Quaresma, Predrag Janičić, Jelena Tomašević,
Milena Vujošević-Janičić, and Dušan Tošić

Abstract

With a large number of tools focusing on visualising geometrical constructions or on proving properties of constructed objects (or both), there is an emerging need of linking them, and making them and their corpora, widely usable. A common setting that links these tools would be important in the field of geometrical constructions and in their role in education. In the following text we propose a common, XML-based, interchange format for descriptions of geometrical constructions and proofs. We also present a XML library providing support for dynamic geometry software and automatic theorem provers, and its integration into our web-based GeoThms system.

Contents

1 Introduction

Dynamic geometry software (DGS), such as *Cinderella*, *Geometer's Sketchpad*, *Cabri*,[1] visualise geometric objects and link formal, axiomatic nature of geometry, most often Euclidean, with its Cartesian models and corresponding illustrations. The common experience tell us that, dynamic geometry tools significantly help students to acquire knowledge about geometric objects and, more generally, for acquiring mathematical rigour.

In many DGSs, a geometric construction is specified using, explicitly, a formal language. In others, the construction is made interactively, by clicking specific buttons and/or icons, but behind this approach there is also a formal geometrical language, although usually hidden from the user. All these languages share many primitive commands (related to geometrical constructions), but there are also differences in the set of supported commands, and they follow different syntax rules.

Besides DGSs, there are automated theorem provers (ATP) specialised for geometrical constructions. Some of them aim at producing traditional, human readable geometrical proofs [1, 6, 7, 14, 9].

With a large number of tools focusing on visualising geometrical constructions or on proving properties of constructed objects (or both), there is an emerging need of linking them and making widely usable constructions and proofs generated with different tools. This would help in the progress of the field of geometrical constructions, including their role in education.

We believe that descriptions of geometrical constructions and geometrical proofs should be put into the XML framework, by defining a *normal form*, linked to different formats. In this paper we describe a XML-based system built on such XML-based format. These are some of the most important motivating arguments for using XML in storing descriptions of geometrical constructions and proofs, and as interchange format:

- instead of raw, plain text representation, geometrical constructions will be stored in strictly structured files; these files will be easy to parse, process, and convert into different forms and formats;
- input/output tasks will be supported by generic, external tools and different geometry tools will communicate easily;
- growing corpora of geometrical constructions will be unified and accessible to users of different geometry tools;
- easier communication and exchange of material with the rest of mathematical and computer science community;
- there is a wide and growing support for XML;
- different sorts of presentation (text form, LaTeX form, HTML) easily enabled;
- strict content validation of documents with respect to given restrictions.

[1] See `http://www.cinderella.de`, `http://www.keypress.com/sketchpad/`, `http://www.cabri.com`

We have implemented converters for two DGSs, confirming, in this way, that the proposed XML format can serve its main purpose. We have also developed XML support for automatically generated proofs of constructive geometrical theorems. These tools, together with rendering tools (tools for visual presentation of XML files) were incorporated in our GeoThms framework.

2 Background

In this section we give some basic background information about geometrical constructions, XML, and our GeoThms framework that links DGSs, ATPs, and a repository of geometry problems.

2.1 Geometrical Constructions

For hundreds, or even thousands of years geometric construction problems have been one of the most attractive parts of geometry and mathematics. A geometric construction is a sequence of specific, primitive construction steps. These primitive construction steps (also called *elementary constructions*) are based on using a *ruler* (or a *straightedge*[2]) and a *compass*, and they are:

- construction (with *ruler*) of a line such that two given points belong to it;
- construction (with *ruler*) of a segment connecting two points;
- construction (with *compass*) of a circle such that its centre is one given point and such that the second given point belongs to it;
- construction of a point which is an intersection of two lines (if such a point exists);
- construction of intersections between a given line and a given circle (if such points exist).

By using the set of primitive constructions, one can define more complex constructions (e.g., the construction of a right angle, a construction of the midpoint of a line segment, etc.).

The abstract (i.e., formal, axiomatic) nature of geometric objects have to be distinguished from their usual interpretations. A geometric construction is a procedure consisting of abstract steps and it is not a picture, but for each construction there is its counterpart in the standard Cartesian model.

2.2 XML

Extensible Markup Language (XML) is a simple, very flexible text format for data structuring using tags, inspired by SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the Web and elsewhere[3]. It is called extensible because it is not a fixed format like HTML

[2]The term "straightedge" is sometimes used instead of "ruler" in order to emphasise there are no markings which could be used to make measurements.

[3]`http://www.w3.org/XML/`

(a single, predefined markup language), instead the tags indicate the semantic structure of the data, rather than (only) its layout in a browser. XML is actually a "metalanguage", — a language for describing other languages, which lets one design his/her own customised markup languages for limitless different types of documents. XML provides a structured way of transmitting information between programs and systems. It is intended to make it easy to define document types, to write and maintain documents, and to share them across the Internet.

However, XML is not just for Web pages: it can be used to store any kind of structured information, and to enclose or encapsulate information in order to pass it between different computing systems. An XML document can carry both presentation (i.e., plausible visualisation) and content information. XML is a project of the World Wide Web Consortium (W3C) and is a public format — it is not a proprietary development of any company. Almost all browsers that are currently in use support XML natively.

Data type definitions (DTDs) provide a formal specification of the constraints on the structure of data presented in XML form. A DTD is given as a formal description in XML declaration syntax. It sets out what names are to be used for the different types of element, where they may occur, and how they all fit together. This formal description enables automatic verification ("validation") of whether a document meets the given syntactical restrictions. This way, groups sharing data of a similar sort can agree on their XML representation and corresponding DTDs.

Extensible stylesheet language transformation (XSLT) is a document processing language that is used to transform the input XML documents to output files. An XSLT style-sheet declares a set of rules (templates) for an XSLT processor to use when interpreting the contents of an input XML document. These rules tell to the XSLT processor how that data should be presented: as an XML document, as an HTML document, as plain text, or in some other form.

Scalable Vector Graphics (SVG) is a language, based on XML, for describing two-dimensional graphics and graphical applications. As for other XML applications, there is a W3C recommendation for SVG[4].

2.3 GeoThms Framework

GeoThms[5], is a Web workbench in the field of constructive problems in Euclidean geometry. It is a framework that links dynamic geometry software, geometry automatic theorem provers, and a repository of geometry problems (geoDB), providing a common web interface for all these tools. Its tight integration of dynamic geometry tools and automatic theorem provers and its repository of theorems, figures and proofs, gives the user the possibility to easily browse through the list of geometric problems, their statements, illustrations and proofs. Currently, there are the following tools integrated in GeoThms:

[4] `http://www.w3.org/Graphics/SVG/`

[5] GeoThms is accessible from `http://hilbert.mat.uc.pt/~geothms`.

[6] GCLC package is freely available from `www.matf.bg.ac.yu/~janicic/gclc/`. There are versions for Windows and Linux.

[7] Eukleides is available from `http://www.eukleides.org`. The first author of this paper is responsible for the Portuguese version of Eukleides: EukleidesPT is available from `http://gentzen.mat.uc.pt/~EukleidesPT/`

GCLC [5][6] and Eukleides [8, 10][7] are two DGS; they both use (similar) geometry drawing languages in which producing mathematical illustrations is based on "describing figures", rather than on "drawing figures". These descriptions directly reflect meaning of mathematical objects to be presented, and are easily understandable to mathematicians. Both tools have graphical user interfaces and the ability to produce LaTeX files with illustrations for geometrical constructions.

GCLCprover is an ATP based on the area method [2, 3, 6, 9], a theorem prover that allows formal deductive reasoning about objects constructed with the help of DGSs. It produces proofs that are human-readable, and with a clear justification for every proof step. GCLCprover is tightly integrated with the GCLC, which means that one can use the prover to reason about a GCLC construction, without changing and adapting it for the deduction process. The users only need to add a statement that they want to prove. The geometrical constructions made within GCLC are internally transformed into primitive constructions of the area method, and in certain cases, some auxiliary points are introduced. With support of our XML library, it is also possible to reason about the Eukleides constructions.

geoDB database gives support to the other tools, keeping the information, and allowing for its fast retrieving whenever necessary. Constructions are described and stored in XML form. Figures are generated from the XML files, by DGSs, and stored in suitable formats (JPEG and SVG). Conjectures are described and stored in a form that extend geometric specifications. The specifications of conjectures are used via converters by ATPs. Proofs are generated by ATPs and stored in suitable formats (PDF and XML in compressed form).

3 Overall Architecture

In this section we provide some motivating arguments for introducing XML-based format in representing geometrical constructions and geometrical proofs. Also, we propose the architecture of a system based on these motivations and ideas (the actual implementation of our system is described in the next section).

3.1 Representation of Construction Descriptions

All dynamic geometry tools use some formal languages for describing geometrical objects (either a hidden, underlying language or a user-oriented language). Consider, for instance, two equivalent descriptions (in GCLC language and in Eukleides language) of the same construction given in Figure 1. GCLC language and Eukleides language were developed/defined independently by independent authors. Corresponding descriptions in languages of many other geometry tools are similar. The reason for this is that all these tools describe (standardised) elementary constructions by ruler and compass (see 2.1) and deal with similar additional requests for drawing and labelling geometrical figures. So, all of these languages are very similar, but still different (due to different main purposes, different authors, different implementations, etc.)

```
dim 80 80                          frame(0,0,8,8)
point A 10 30                      A = point(1,3)
point B 60 10                      B = point(6,1)
point C 50 70                      C = point(5,7)

med a B C                          a = bisector(segment(B,C))
med b A C                          b = bisector(segment(A,C))
med c B A                          c = bisector(segment(B,A))
intersec O_1 a b                   O1 = intersection(a,b)
intersec O_2 a c                   O2 = intersection(a,c)

drawline a                         draw(a)
drawline b                         draw(b)
drawline c                         draw(c)
drawsegment A B                    draw(segment(A,B))
drawsegment A C                    draw(segment(A,C))
drawsegment B C                    draw(segment(B,C))

cmark_lb A                         draw(A); label(A,-90:)
cmark_b B                          draw(B); label(B,-90:)
cmark_t C                          draw(C); label(C,90:)
cmark_t O_1                        draw(O1); label(O1,90:)
cmark_lb O_2                       draw(O2); label(O2,-135:)

drawcircle O_1 A                   draw(circle(O1,length(segment(O1,A))))
```

Figure 1: Equivalent descriptions of a construction in GCLC (left) and in Eukleides (right) languages

In order to enable communication between these tools and converting files between different formats, it is good to have a single target format, a format that could define a common normal form for different tools. We propose one such format, within a general XML specification. Figure 2 shows how the description given in Figure 1 can look in XML version. Notice, again, a direct link between the XML representation and representations in GCLC and Eukleides languages.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE figure SYSTEM "GeoCons.dtd">
<?xml-stylesheet href="GeoConsHTML.xsl" type="text/xsl"?>

<figure>
<draw>
<dimensions width="80.000000" height="80.000000"></dimensions>
</draw>
<define>
<fixed_point x="10.000000" y="30.000000">A</fixed_point>
<fixed_point x="60.000000" y="10.000000">B</fixed_point>
<fixed_point x="50.000000" y="70.000000">C</fixed_point>
</define>
<construct>
<segment_bisector><new_line>a</new_line><point>B</point><point>C</point></segment_bisector>
<segment_bisector><new_line>b</new_line><point>A</point><point>C</point></segment_bisector>
<segment_bisector><new_line>c</new_line><point>B</point><point>A</point></segment_bisector>
<intersection><new_point>O_1</new_point><line>a</line><line>b</line></intersection>
<intersection><new_point>O_2</new_point><line>a</line><line>c</line></intersection>
</construct>
<draw>
<line>a</line>
<line>b</line>
<line>c</line>
<segment><point>A</point><point>B</point></segment>
<segment><point>A</point><point>C</point></segment>
<segment><point>B</point><point>C</point></segment>
</draw>
</figure>
```

Figure 2: XML version of construction descriptions given in Figure 1

Converting from a DGS language to XML, would be performed by a specific converter, naturally relying on the DGS's parsing mechanism. Converting from XML to a DGS language, will be implemented via a XSLT file.

Having converters from, and to, XML format for all DGSs, we (indirectly) have converters from each format to any other format. Thus, in this way, the

base for a common interchange format is provided. XML is a natural framework for such interchange format, because of its strict syntax, verification mechanisms, suitable usage on the Internet, and a large number of available supporting tools.

XML descriptions of constructions can be, by means of XSLT, also transformed into HTML format that is convenient for human-readable display in browsers. It can also be transformed into different representations, such as natural language form.

A specific DTD document would define syntactical restrictions for construction descriptions. This DTD document can then be used, in conjunction with the generic XML validation mechanism (based on DTSs or XML schemes), for verifying whether a given description of a geometrical construction is legal.

3.2 SVG support

As said in the previous subsection, XML format can be used for representing descriptions of geometrical constructions and, hence, as an interchange format for different geometry tools. On the other hand, this representation can be used for visualisation of constructions, by using SVG. The visualisation data (in SVG) can be generated directly from the XML description of a construction (basically requiring a new geometry tool). Another possibility is to implement, within a DGS, an *export to* SVG option.

With this option implemented, the visualisation data (in SVG) could be generated from the XML description indirectly — the XML description would be first converted to a representation of the geometry tool, and then further to SVG format. Note that, with only a limited numbers of converters, a wide range of processing of geometrical descriptions would be possible.

3.3 Representation of Proofs

Geometrical proofs should be stored in a way that provides:

- strict verification;
- different sorts of presentation for easier understanding.

Geometrical proofs could be stored in different forms, for instance in axiomatic form (e.g., in Hilbert-style, sequent calculus style, etc). Representing higher-level proofs, produced by the area-method, is also interesting. Proofs generated by this method consist of sequences of equalities involving expressions over geometry quantities (such as a signed area of triangle). For each step of the proof, leading from one equality to another, there is a detailed justification (in terms of used definition or lemma): for elimination steps, geometrical simplification steps, and for algebraic simplification steps. These proofs have linear structure, but may involve subproofs (proofs of lemmas).

4 Implementation

In this section we describe our XML suite for geometrical constructions and geometrical proofs. It follows motivations and ideas given in Section 3 and consists of:

- newly defined XML-based format for representing geometrical constructions with corresponding DTD; this format covers standard constructions by ruler and compass, but also a range of other devices supported by dynamic geometry tools (including, for instance, compound constructions, transformations, labelling etc.);
- converters from dynamic geometry tools to XML-based form (currently, there are converters for GCLC and Eukleides; these converters were written in the programming languages C++ and C, as the main tools themselves);
- converters for descriptions of constructions from XML-based form to dynamic geometry tools (currently, there are converters for GCLC and Eukleides; these converters were implemented as XSLT files);
- a converter for descriptions of constructions from XML-based form to a simple, readable HTML form (with syntax colouring features, provided for better readability); this converter was implemented as a XSLT file;
- a converter from XML-based form to a natural language form (currently, only for English language); this converter was implemented as a XSLT file;
- a tool for exporting figures from dynamic geometry tools to SVG format (currently, there is a converter for GCLC; this converter was written in the programming language C++, as the main tool itself);
- newly defined XML-based format for representing proofs of properties of geometrical constructions with a corresponding DTD; the format is adapted for the area-method;
- a tool for exporting proofs from automated theorem provers systems to XML-based form (currently, there is a converter for GCLCprover; this converter was written in the programming language C++, as the main tool itself);
- a converter for proofs from XML-based form to a simple, readable HTML form (with syntax colouring features, and other features for better readability); this converter was implemented as a XSLT file;

GeoThms uses XML format to high extent, for storing, communicating, and presenting data. The presented suite is available:

- via GeoThms (from `http://hilbert.mat.uc.pt/~geothms`)
- and partly within a distribution package for GCLC (from `http://www.matf.bg.ac.yu/~janicic/gclc/`).

Some examples built with the help of the above mentioned tools are given in Section 5.

5 Examples

Figure 2 shows the XML code that corresponds to construction descriptions given in Figure 1. The code is simple and readable. Within the code, there were points A, B, C introduced, and then the bisectors a and b of the sides BC and AC were constructed. The intersection of a and b is denoted by O_1 (note that this point is the centre of circumcircle of the triangle ABC). The four points, the three sides of the triangles, and the circle with the centre O_1, containing the point A are shown.

```
<!--**************constructions**************-->
<!ELEMENT construct (intersection|intersection_cc|intersection_cl|midpoint|
                foot|random_point_on_line|translate|towards|rotate|half_turn|
                line_reflection|inversion|ruler|parallel|perpendicular|
                segment_bisector|angle_bisector|compass)*>
<!ELEMENT new_point (#PCDATA)>
<!ELEMENT intersection (new_point,line,line)>
<!ELEMENT intersection_cc (new_point,new_point,circle,circle)>
<!ELEMENT intersection_cl (new_point,new_point,circle,line)>
<!ELEMENT midpoint (new_point,point,point)>
<!ELEMENT foot (new_point,point,line)>
<!ELEMENT random_point_on_line (new_point,point,point)>
<!ELEMENT translate (new_point,vector,point)>
<!ELEMENT towards (new_point,vector,coefficient)>
<!ELEMENT rotate (new_point,center,angle,point)>
<!ELEMENT half_turn (new_point,center,point)>
<!ELEMENT line_reflection (new_point,line,point)>
<!ELEMENT inversion (new_point,circle,point)>
```

Figure 3: Fragment of the DTD for geometrical constructions

The contents of the file shown in Figure 2 is valid with respect to a special-purpose DTD, developed for geometrical constructions. Part of this DTD is shown in Figure 3.

The contents of the file shown in Figure 2 was generated by the converter from GCLC to XML format.

The contents of the file shown in Figure 2, transformed by the XSLT files `geocons-gclc.xsl` and `geocons-eukleides.xsl`, gives (exactly) the contents in GCLC and *Eukleides* format shown in Figure 1.

The contents of the file shown in Figure 2, transformed by the XSLT file `geoconsHTML.xsl` gives simple and readable description of the construction presented in HTML. The contents of the file shown in Figure 2, transformed by the XSLT file `geoconsNL.xsl` gives a similar description, in HTML, but in a natural-language form (Figure 4).

A SVG-based visualisation of the construction given in Figure 2 can be obtained by first converting the file shown in Figure 2 to GCLC format, and then by using the option for exporting from GCLC format to SVG.

Our XML suit also has support for storing and presenting geometrical proofs. The current support is aimed only at the proofs produced by the area method (but it is subject to changes and extensions for other proof styles). Consider the construction described in Figure 2. If we construct a bisector c of the side AB, and if we construct the intersection O_2 of the lines a and c, then the points O_1 and O_2 will be identical. This property can be proved by GCLCprover. Figure 5 shows part of this proof presented in HTML. The XML code generated by GCLCprover is simple and readable. It is valid with respect to a DTD developed

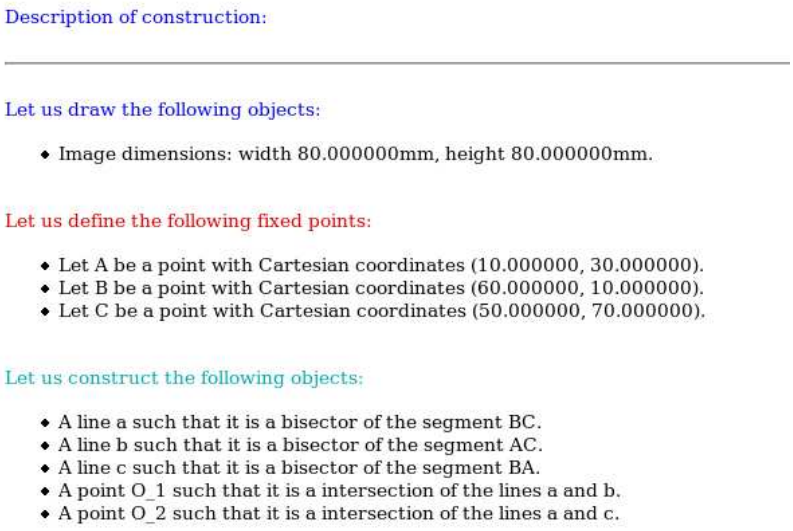

Figure 4: Fragment of a natural-language rendering of the XML file shown in Figure 2

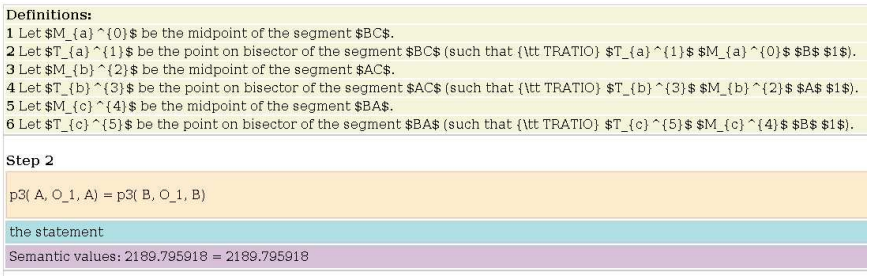

Figure 5: A fragment of a proof generated by GCLCprover

```
<!--******** Definitions **************-->
<!ELEMENT definitions (definition)*>
<!ELEMENT definition (#PCDATA)>

<!--******** Proof **************-->
<!ELEMENT proof (proof_step|lemma)*>
<!ELEMENT proof_step (equality,explanation,semantics)>

<!ELEMENT lemma (proof,status)>
<!ATTLIST lemma level CDATA #REQUIRED>

<!ELEMENT equality (expression,expression)>
<!ELEMENT inequality (expression,expression)>

<!ELEMENT expression (number|constant|sum|mult|fraction|segment_ratio|signed_area3|
signed_area4|pythagoras_difference3|pythagoras_difference4)>
```

Figure 6: A fragment of the DTD for proofs of properties of geometrical constructions

for proofs of properties of geometrical constructions. Part of this DTD is shown in Figure 6.

6 Conclusions and Further Work

We have presented a case for using XML in describing geometrical constructions and proofs, and as an interchange format for dynamic geometry tools. We gave a brief description of the notion of geometrical constructions, XML, and the geometrical software tools that already use our support for XML. Our XML suite is publicly available and used in the GeoThms framework.

This work is related to work in other domains of automated reasoning: joint efforts of numbers of researchers led to standards such as DIMACS (for propositional logic) [4] and SMT (for satisfiability modulo theory) [11] and repositories of problems such as SAT-lib (for propositional logic) [12], TPTP (for predicate logic) [13], SMT-lib (for satisfiability modulo theory) [11] etc. Such efforts, standards, and libraries are very fruitful for easier exchange of problems, proofs, and even program code, and they help advancing the underlying field.

We are planning to work on further improvements (based on XML schemes) of the validation mechanism including some semantics checks. Also, we will work on extending and improving the format for proofs, and especially on using applications such as MathML, and schemes for describing mathematical contents such as OMDoc.

We intend to further build the database of geometrical constructions within GeoThms and, hopefully lead it to a major public resource for geometrical constructions, linking a number of geometry tools, format and repositories.

References

[1] Bruno Buchberger and et. al. Theorema: Towards computer-aided mathematical theory exploration. *Journal of Applied Logic*, 2006.

[2] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated production of traditional proofs for constructive geometry theorems. In Moshe Vardi, editor, *Proceedings of the Eighth Annual IEEE Symposium on Logic in Computer Science LICS*, pages 48–56. IEEE Computer Society Press, June 1993.

[3] Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. Automated generation of readable proofs with geometric invariants, I. multiple and shortest proof generation. *Journal of Automated Reasoning*, 17:325–347, 1996.

[4] DIMACS. Satisfiability suggested format. ftp://dimacs.rutgers.edu/pub/challenge/satisfiability/doc/satformat.tex.

[5] Predrag Janičić. GCLC – a tool for constructive euclidean geometry and more than that. In Nobuki Takayama, Andres Iglesias, and Jaime Gutierrez, editors, *Proceedings of International Congress of Mathematical Software (ICMS 2006)*, number 4151 in LNAI. Springer-Verlag, 2006.

[6] Predrag Janičić and Pedro Quaresma. System description: GCLCprover + geothms. In Ulrich Furbach and Natarajan Shankar, editors, *IJCAR 2006*, LNAI, pages 145–150. Springer-Verlag, 2006.

[7] Julien Narboux. A decision procedure for geometry in coq. In *Proceedings TPHOLS 2004*, volume 3223 of *Lecture Notes in Computer Science*. Springer, 2004.

[8] Christian Obrecht. Eukleides. http://www.eukleides.org/.

[9] Pedro Quaresma and Predrag Janicic. Framework for constructive geometry (based on the area method). Technical Report 2006/001, Centre for Informatics and Systems of the University of Coimbra, 2006.

[10] Pedro Quaresma and Ana Pereira. Visualização de construções geomtricas. *Gazeta de Matemtica*, (151):39–41, Junho 2006.

[11] Silvio Ranise and Cesare Tinelli. The SMT-LIB format: An initial proposal. http://goedel.cs.uiowa.edu/smt-lib/, 2003.

[12] *Satlib: An online resource for research on sat*. IOS Press, 2000. SATLIB is available online at www.satlib.org.

[13] Geoff Sutcliffe. The TPTP problem library. http://www.cs.miami.edu/~tptp/TPTP/TR/TPTPTR.shtml.

[14] Zheng Ye and et. al. Geometry expert. http://woody.cs.wichita.edu/gex, 2004.