Geometry Construction Recognition by the Use of Semantic Graphs

Pedro Quaresma
pedro@mat.uc.pt

Departamento de Matemática
Faculdade de Ciências e Tecnologia
Universidade de Coimbra
3001-501 COIMBRA, PORTUGAL

Yannis Haralambous
yannis.haralambous@telecom-bretagne.eu

Département Informatique, Télécom Bretagne
UMR CNRS 6285 Lab-STICC
Technopôle Brest-Iroise, CS 83818
29238 BREST Cedex 3, FRANCE

Abstract

Given the large number of dynamic geometry systems (DGS), geometry automated theorem provers (GATP) and repositories of geometric information (constructions and/or conjectures), we face the need of a query mechanism for formal descriptions of geometric constructions. The DGS and GATP describe the geometric constructions using formal languages where the elements and the relations between them are described formally and not in terms of a given geometric model. Given a formal language we need to be able to look for similar construction, sub-constructions or even different construction sharing a common property, e.g., a set of constructions about right angled triangles. Our approach is to transform the geometric construction into a semantic graph representation of the construction, in a given ontology. Graph pattern recognition algorithms can then be used to search for the similarities we need and the results brought back to the geometric setting.

1 Introduction

Dynamic geometry systems (DGS) [11] distinguish themselves from drawing programs in two major ways, the first is their knowledge of geometry. Indeed, from a initial set of objects drawn freely in the Cartesian plane (or maybe, on another model of geometry), one can specify/construct a given geometric figure using relations between the objects, e.g., the intersection of two non-parallel lines, a line perpendicular to a given line and containing a given point, etc. That is, one uses a DGS by constructing a geometric figure with geometric objects and geometric relations between them, not placing points on specific cartesian coordinates. Another major feature of a DGS is its capability to introduce dynamics to a given geometric construction. Given the fact that one specifies/constructs a geometric figure using a set of basic elements, e.g. points, lines, circles and relations between then, the DGS allow its user to move one of the basic (free) elements form its initial placement to another placement in the Cartesian plane, the relations will be kept, so a movement in a single point can entail the movement of almost all the other elements in the construction, i.e., when moving a basic object, we will move that object and all the other elements that are related to it, always preserving the geometric properties of the construction.

Most (if not all) DGS possess a formal language for the specification of geometric constructions. In some systems this formal language is explicit, in others it is hidden from the user by the graphical interface. The *intergeo* project designed a common format, called I2G, for this formal language which is already accepted by many DGSs [1, 9].

Geometry automated theorem provers (GATP), being formal systems, need a formal language to describe geometric conjectures. GATPs are nowadays mature tools capable of proving hundreds of geometric conjectures [3]. There are two major lines of research: the synthetic proof style and the algebraic proof style. The algebraic proof style begins by reducing the geometric properties to algebraic properties expressed in terms of Cartesian coordinates, proving the theorem by pure algebraic methods, so they do not belong to the realm of the geometric reasoning. The synthetic proof style GATP uses geometric reasoning, and its formal language is an extension of the formal language used by the DGS. The I2GATP project goal is to define a common language, an extension of the I2G language, to the DGS/GATP tools [7].

The design of common languages, and the emergence of Web repositories of geometric knowledge is an attempt to make widely available the already vast data set of geometric knowledge. The *intergeo* project [5], the *GeoThms* [8] and the *TGTP* [6] systems already meet some of these goals having provided a large data set of geometric information widely available. In these systems the question of querying the geometric construction is not solved, that is, it is not possible to query the data set for a construction similar to some other construction, or to query for all constructions having some common geometric properties.

The goal of our research is to develop a search mechanism for geometric constructions (done by a DGS or a GATP) using the formal specification of the construction. Our approach is to transform the geometric construction into a semantic graph representation of its elements and relations, in a given ontology. The resulting semantic graphs can then be used, using graph pattern recognition algorithms, to detect matching constructions, and the results can be presented to the user once converted back to their geometric representation.

1.1 State of the Art

The term “Geometric Pattern Matching” refers to the recognition of shapes in a given set of points and it is a important area of research with applications in computer image processing, manufacturing, robotics, VLSI design, military intelligence, etc. with many articles written under many different approaches, but that it is not the goal we pursue. By geometry construction recognition we mean the search for pattern not on a given model of geometry but in the formal specification of the construction, or, *equivalently*, in its semantic graph representation, for a given ontology. As far as we known this is an area still to be explore, we are making the first steps in that direction.

2 Geometric Constructions and their Semantic Graph Representations

The DGS and most of the GATP are based in a constructive geometry [10]. A construction is specified stating an initial set of points, the free points, implicitly universally quantified over a field of characteristic different from two, and from these initial elements, using a set of constructive rules the geometric construction is built [2, 4]. For rendering the DGS will attach to the free points some Cartesian coordinates, but these coordinates will be ignored by the GATP.

Let us consider a (very) small example (see Fig. 1).

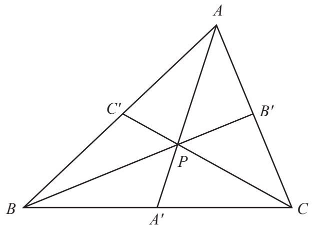


Figure 1: Geometric construction.

Points A, B and C are the free objects of this construction, starting with them we define new elements A', B' and C' as being the middles of the opposite line segments, and P as the intersection of AA', BB', CC' and hence the barycenter of the triangle. The *GeoThms* and the *TGTP* systems share a database with more them 180 constructions like this one, specified in a constructive geometry language, the geometric queries are a much needed feature for those systems.

To represent this construction we need to chose an ontology. Most ontologies will contain the concept of "point," but, depending on the theorem to prove, one may add concepts for "line," "line segment," "angle," "length," etc. Typical relations are "belongs to," "is parallel to," etc. In our case, since points A', B' and C' are defined as being middles of line segments, we need the relation "is middle of," which is a superrelation of "belongs to," in the partial hierarchical order of the ontology.

In Fig. 2 one can see the semantic graph of construction 1 in the ontology using points (represented by □) and line segments (repr. by △) as concepts, and relations "belongs to" (→) and "is middle of" (dashed →).

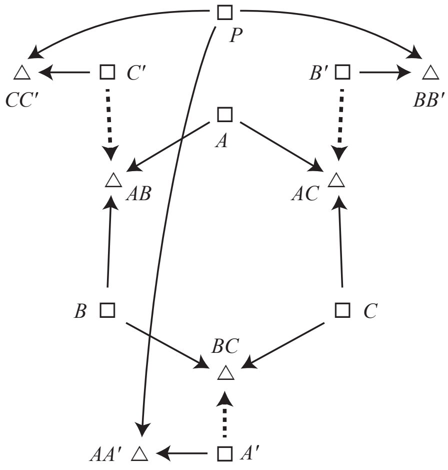


Figure 2: Semantic graph of construction 1. (Labels have been placed only to allow the reader to establish a connection with the original geometric figure — they are not part of the semantic graph per se.)

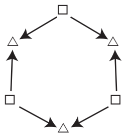
Fig. 2 represents construction 1 entirely, in the given ontology. We could also represent the same construction in an ontology with relation "has same length" instead of "is middle of;" in that case we would use "has same length" arrows between nodes AC' and BC' in both directions, instead of the "is middle of" arrow between C' and AB.

Changing ontologies will cause a transformation of semantic graphs, and this is an operation that can be done mechanically.

3 Querying Geometric Constructions

Every user query can be transformed in a pair consisting of a (minimal) ontology and a (small) semantic graph. For example, the query *find a construction containing a triangle* would result into the ontology {"point" (□), "line segment" (△), "belongs to" (→)} and the semantic graph

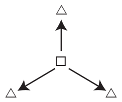
$G =$



By representing all constructions in this ontology (which, in the case of Fig. 2, would require to replace the relation "is middle of" by the weaker relation "belongs to"), we only need to seek for the presence of G as a semantic subgraph of the representation of each construction.

Here is another example: to *find all constructions containing an intersection of three line segments* (or *three lines*, in a slightly different ontology), one would need to seek the presence of the following subgraph:

$G' =$



4 Conclusions and Future Work

Having already defined a semantic graph representation counterpart for the GC geometric constructions and a way of extracting information from it we need to define a query language that can be used in the Intergeo, GeoThms and TGTP (and others) systems, i.e., we need a user interface enabling users to make queries about geometric constructions and/or properties. The support for the geometric languages (I2G and I2GATP) and the trimming of the search mechanism to be used on the large set of geometric construction is also important.

After a query has been submitted to the engine, it will need to find the (minimal) ontology needed, and, if necessary, convert constructions into it. Then it will detect semantic subgraphs and either display them to the user, or display the corresponding parts of the geometric constructions, or simply reply with a boolean value for constructions matching the query.

An interesting research direction to pursue in the future, is also to investigate how inference used in proofs by GATPs, interacts with the semantic representation.

References

[1] The Intergeo Consortium. Intergeo implementation table. `http://i2geo.net/xwiki/bin/view/I2GFormat/ImplementationsTable`, 2012.

[2] Predrag Janičić. Geometry constructions language. *J. Autom. Reasoning*, 44(1-2):3–24, 2010.

[3] Predrag Janičić and Pedro Quaresma. System description: GCLCprover + GeoThms. In Ulrich Furbach and Natarajan Shankar, editors, *Automated Reasoning*, volume 4130 of *Lecture Notes in Computer Science*, pages 145–150. Springer Berlin / Heidelberg, 2006. ISBN 3-540-37187-7. doi: 10.1007/11814771_13.

[4] Predrag Janičić, Julien Narboux, and Pedro Quaresma. The Area Method: a recapitulation. *Journal of Automated Reasoning*, 48(4): Springer, 2012. doi: 10.1007/s10817-010-9209-7.

[5] U. Kortenkamp, C. Dohrmann, Y. Kreis, C. Dording, P. Libbrecht, and C. Mercat. Using the Intergeo platform for teaching and research. In *Proceedings of the 9th International Conference on Technology in Mathematics Teaching (ICTMT-9)*, 2009.

[6] Pedro Quaresma. Thousands of Geometric problems for geometric Theorem Provers (TGTP). In Pascal Schreck, Julien Narboux, and Jürgen Richter-Gebert, editors, *Automated Deduction in Geometry*, volume 6877 of *Lecture Notes in Computer Science*, pages 169–181. Springer Berlin / Heidelberg, 2011. ISBN 978-3-642-25069-9. doi: 10.1007/978-3-642-25070-5_10.

[7] Pedro Quaresma. An XML-format for conjectures in geometry, (work-in-progress). In *Proceedings of the Conferences on Intelligent Computer Mathematics, CICM 2012, Work-in-Progress*. Jacobs University, Bremen, Germany, 8-13 July 2012. URL `http://arxiv.org/abs/1207.2511`.

[8] Pedro Quaresma and Predrag Janičić. GeoThms – a Web System for euclidean constructive geometry. *Electronic Notes in Theoretical Computer Science*, 174(2):35 – 48, 2007. doi: 10.1016/j.entcs.2006.09.020.

[9] E. Santiago, Maxim Hendriks, Yves Kreis, Ulrich Kortenkamp, and Daniel Marquès. I2G Common File Format Final Version. Technical Report D3.10, The Intergeo Consortium, 2010. URL `http://i2geo.net/xwiki/bin/view/I2GFormat/`.

[10] Jan von Plato. The axioms of constructive geometry. In *Annals of Pure and Applied Logic*, volume 76, pages 169–200, 1995.

[11] Wikipedia. List of interactive geometry software. `http://en.wikipedia.org/wiki/List_of_interactive_geometry_software`, August 2012.