Automated Generation of High School Geometric Questions Involving Implicit Construction

Rahul Singhal[1], Martin Henz[1] and Kevin McGee[2]
[1]*School of computing, National University of Singapore, Singapore, Singapore*
[2]*Department of Commmunications and New Media, National University of Singapore, Singapore, Singapore*

Keywords: Automated Deduction, Constructive Geometry, Graph-based Knowledge Representation.

Abstract: We describe a framework that combines a combinatorial approach and automated deduction to generate geometry problems which require implicit constructions for their solution. This is an extension of our framework developed for generating geometric questions without construction. Such a system would help teachers to quickly generate large numbers of questions involving implicit construction on a geometry topic. Students can explore, revise and master specific topics covered in classes and textbooks based on construction-based generated questions. This system may also help standardize tests such as GMAT and SAT. Our novel methodology uses (i) a combinatorial approach for generating geometric figures and objects for construction (ii) automated deduction to generate new questions and solutions. By combining these methods, we are able to generate questions involving finding or proving relationships between geometric objects based on a specification of the geometry objects, concepts, theorems and construction object to be covered by the questions. Experimental results show that a large number of questions can be generated in a short time.

1 INTRODUCTION

Construction-based Euclidean geometry questions refers to geometric questions which require constructions, namely, to find proofs with additional lines, points, or arcs constructed by a compass and a ruler. Finding a construction is a hard task even for human problem solvers. Since one can draw many segments and arcs at any point of a proof, the search space is enormous. Hence generating construction-based questions on a user-selected geometric topics is a more challenging task.

It is important to understand construction-based geometry when studying a course because it does not follow any set pattern. In Euclidean geometry, one can only learn the axioms and results proven from these axioms. The student must apply these axioms with no set pattern or list of steps for solving such questions. Therefore, a question may have (possibly infinitely) many solutions. To practice the required problem solving skills, students require a large number of different types of geometry questions on various concepts which requires implicit construction of various geometric objects. Generally, textbooks and online sites provide a limited predefined number of such type of questions for each topic. Once practiced, these questions lose their purpose of enhancing student thinking. The tedious and error-prone task of generating high-quality questions challenges the resources of teachers. Hence, there is a need for software which assists both teachers and students to generate geometry questions and solutions.

Apart from helping users, the framework of generating questions has scientific contributions to other research areas, such as Intelligent tutor systems (ITS) and Massive Online Open Courses (MOOC).

Various research has been performed in automated deduction of theorems at high school level in the geometry domain, although none with the goal of automatic construction-based question generation. Instead, they mainly demand users to generate the question with the help of tools. In addition, they mainly focus on solving and assessment of the questions. Our survey shows that the currently available geometry systems, such as JGEX, Geogebra, Cinderella and Sketchpad, are not able to automatically generate questions of user specified geometry topics. The closest software is GRAMY tool which can solve questions involving implicit constructions. However it is unable to generate such questions.

The aim of this paper is to develop a framework that can be used to generate construction-based geom-

Singhal R., Henz M. and McGee K..
Automated Generation of High School Geometric Questions Involving Implicit Construction.
DOI: 10.5220/0004947904670472
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 467-472
ISBN: 978-989-758-020-8
Copyright © 2014 SCITEPRESS (Science and Technology Publications, Lda.)

etry questions based on specific inputs, such as geometry objects, theorems and construction objects to be involved in their solution. For a given set of geometry objects, the algorithm can generate a large set of questions along with their solutions. The solutions will involve user desired theorems directly or indirectly. Hence the framework can generate questions to test the theorem on various geometry objects and concepts.

Generated questions from our framework having implicit construction may involve algebraic computations for a solution. Currently, we restrict the relationships between quantitative entities to linear ones. Our framework has a predefined database of theorems, concepts and construction objects which can be used for generating questions. Given a set of user-selected construction objects, our system can generate all possible questions using an existing database of concepts and theorems.

The main contributions of this paper are as follows:

1. Our geometry question generator combines the complementary strengths of a combinatorial approach, pattern matching and deductive reasoning. Construction-based geometry questions can be generated which were not possible previously.
2. A substantial evaluation is provided that demonstrates the effectiveness of our generator. The question generator was able to generate most of the questions, involving implicit construction, covered in the textbooks based on targeted queries.

2 RELATED WORK

There are several automated theorem proving system in the geometric domain such as JGEX (Gao and Lin, 2004), Cinderella and Geometry Explorer (Wilson and Fleuriot, 2006) that allows users to build proofs based on geometric constructions. Angle method (Gao and Lin, 2004), Wu's method and Grobner basis method (Chou et al., 1994) are some of the methods used by the existing geometry systems for automatically proving geometric theorems. However, these methods are either not suitable for question generation or the approach used is not suitable for the high school geometric domain. Hence, the geometric systems are unable to automatically generate geometric questions.

GRAMY (Matsuda and Vanlehn, 2004) is a geometric system that can solve geometric questions which require implicit construction. It uses a forward chaining search followed by a backward step to suggest a construction. GRAMY can only discover proofs that do not involve arithmetic operations. For example, it cannot find proofs that involve inequalities and ratios.

An algorithm developed by Rohit (R. Singh and Rajamani, 2012) used a numerical approach for solving geometric questions involving construction. The algorithm uses the concept of randomness instead of performing symbolic reasoning. Hence, the solutions generated by this algorithm are based on numerical reasoning and out of the scope of high school mathematics.

Our original framework can generate geometric questions based on concepts, theorems and user-selected geometric objects. However, it cannot generate construction based questions. Current work is an extension of the framework developed in (annonymous, 2013).

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

3 GEOMETRY QUESTION GENERATION TASK

Mathematically a geometry question Q generated by our system can be represented by a sextuple (Object O, Concept C, Theorem T, Construction objects CObj, Relationship R, Query type Qt) where:

- O $\in$ (lines, triangles, square, circle, ...)
- C $\in$ (perpendicular, parallel, midpoint, angle-bisector, circumcircle...)
- T $\in$ (Pythagorean theorem, similarity theorem, various triangle-theorems, ...)
- CObj $\in$ (perpendicular, parallel, midpoint, angle-bisector...)
- R $\in$ (syntactic, quantitative)
- Qt $\in$ (syntactic, quantitative)

In order to generate geometry questions, the user has to provide a set of geometry objects O such as triangles, squares, etc., a set of concepts C and a set of construction objects CObj such as perpendicular, parallel, midpoint, etc., which the user would like to cover in the generated question. Optionally the user may select a set of theorems T to be tested by the question. The relationship R can be either syntactic such as perpendicular, parallel, etc., or quantitative such as the giving the length of an object, the ratio of two quantities etc. The query type Qt is the type of generated question that can be asked to find the hidden relationship which can be calculated from the given information. The solution of the generated question would require construction of user-selected

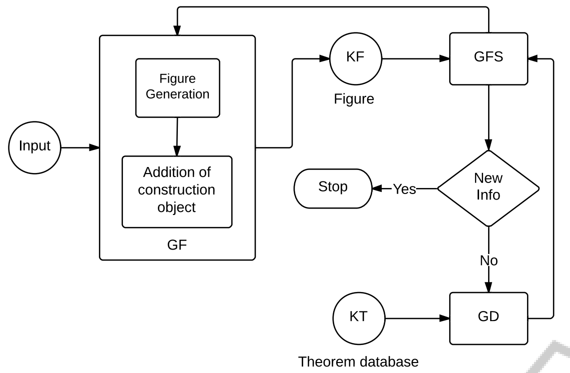

Figure 1: Connection of framework components and knowledge representations.

CObj. Next section describes the framework for generating such questions.

4 QUESTION GENERATION FRAMEWORK

Our framework comprises three major components along with the knowledge databases used for storing geometry figures and a set of predefined theorems. Figure 1 shows the connection of these components. The input consists of geometric objects, concepts, theorems and construction objects selected by the user. The input is fed into the first component, *Generating Figure (GF)*. This component is used for generation of geometric figures from the input. Each figure constitutes a diagrammatic schema (DS) (Greeno et al., 1979) and a set of unknown variables representing the relationship between geometric objects. User-selected construction objects are added to the figure generated by GF according to the algorithm described in the section 4.3. The geometric figure is passed to the second component, *Generating Facts and Solutions (GFS)*. This component is used to find values for the unknown variables representing possible relationships to be covered by the generated question. GFS makes use of the predefined knowledge database of axioms. It results in the formation of a *configuration* (Cfg) containing known values for some relationships between its objects. New information is derived from the configuration Cfg. New information refers to the generation of suitable questions. A question is considered suitable if it covers the essential information such as a new fact and a proper reasoning for the generated fact. A new fact would not be related to the construction object CObj. However, the reasoning of the fact generation would involve the use of CObj.

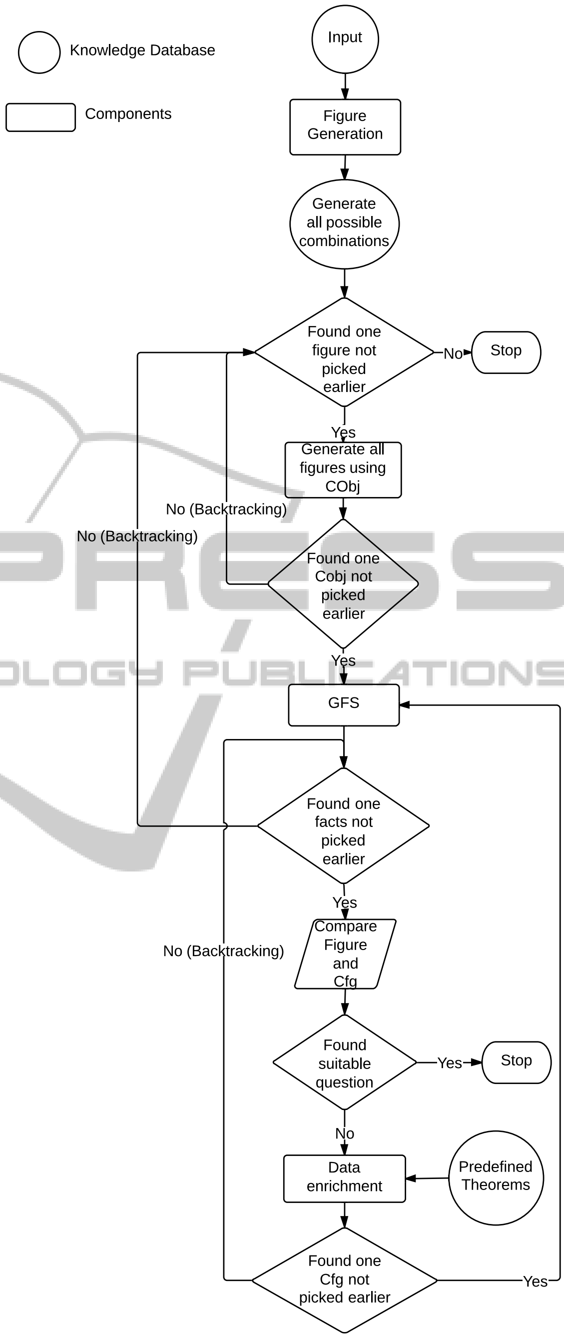

Figure 2: Flow diagram of the algorithm for generating questions.

If the suitability conditions for the generated configuration (Cfg) are not met then the configuration is fed into the last component, *Generating data for the figure (GD)*. GD assigns values to unknown variables of relationships. Repeated processing by GD makes the questions generated from Cfg easier and easier, because the values assigned by GD appear as given facts in the generated questions. GD makes use of

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

a predefined set of theorems and makes sure that the assignment results in successful generation of geometry questions. The configuration generated from this component is again passed to the GFS component and this loop continues until a question is found which meets suitability conditions.

4.1 Algorithm

Figure 2 represents the flowchart describing the algorithm for generating geometric questions involving implicit construction. Algorithm 1 describes the flowchart in further detail.

In this section, Algorithm 1 is explained with the help of a running example. Figure 3 shows the step by step execution of the algorithm. We select the following input in our example.

- Object: triangle and line segment
- Concept: perpendicular
- Construction object: Parallel lines
- Theorem: Pythagorean Theorem
- Number of questions: 1

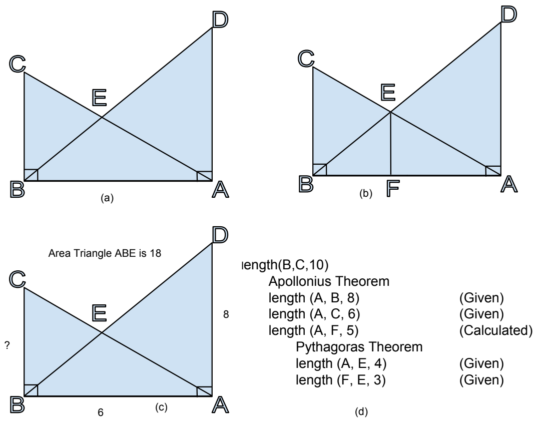

Figure 3: (a) The figure generated by using *triangle* and *perpendicular* as the geometry objects using the GF function. (b) A parallel line is drawn parallel to ”CB” from E meeting ”AB” at ”F”. (c) The data is generated for (b) using the GD component. (d) The new fact not involving EF and its derivation involving EF using the GFS component generated from the figure shown in (c).

In the next subsections, we will describe each framework component and its interaction with the knowledge databases.

Data: User selects object(s), concept(s), construction-object(s), theorem(s) and the number of questions to be generated.

Result: Question with single or multiple solutions.

1. Generate all possible figures consisting of geometry objects using GF function from the given input.
2. Find one figure which has not been picked earlier. If found, go to next step else terminate.
3. Add Cobj using the algorithm mentioned in the section 4.3.
4. If new figure is generated, go to next step else do backtracking.
5. Save this figure using KF knowledge database.
6. Assign values to variables of figure obtained in second step from the predefined knowledge database of axioms through GFS function. Configurations (Cfg) are generated from this step.
7. Find one Cfg not picked earlier. If found, goto next step. If not found, backtracking to step 2.
8. Compare Cfg with the previously stored figure.
9. If the comparison gives the desired suitability then the Cfg is declared as a generated question and the algorithm stops. If the conditions are not met or more number of questions are required, go to the next step.
10. Configuration Cfg obtained from GFS is fed into the third component, GD, to assign more unknown variables. It makes use of KT, a predefined database of theorems.
11. From the configurations obtained from GD, find a new configuration Cfg which has not been chosen earlier. If found, goto step 3. If all configurations have been chosen earlier, goto step 5.

Algorithm 1: Algorithm for generating geometry questions.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

4.2 Generating Figure Configuration from the User Input (GF)

This is the first step executed by Algorithm 1 described in Section 4.1. This component is further divided into two sub-components, *Figure Generation and Addition of construction object*. First sub-component generates a figure through the combination of a predefined number of ways to combine geometric objects. Currently, we are focusing on triangles and line segments. Hence our algorithm includes combinations in which various triangles and lines can intersect. Furthermore, we are currently limiting our algorithm to the intersection of two triangles. Adding more objects may make the figure more complex, possibly leading to unusual questions that are not commonly found in textbooks. However there is no limitation on the intersection of lines and triangles.

Adding Construction Object from the Figure Generated by GF

This component is used for adding geometric objects which will not be shown in the generated question. However, the construction of the object would be required by the user to prove/solve the questions. Generally, more than one construction object can be added for generating questions. However, we have limited our algorithm to addition of single object. Addition of two or more objects may lead to the generation of questions very hard to solve and may not solve the purpose of testing the concepts.

Figure 4 describes the algorithm for adding a construction object in a figure (input) generated by the first sub-component. The algorithm first checks for the user-selected construction objects. If the user has selected an object, the algorithm looks for the preexistence of the selected construction object. In case the object is not present, algorithm adds it in the figure and terminates. The algorithm does nothing if the user-selected object is already present in the figure.

Currently, our algorithm is limited to the construction of a line between two existing points, parallel, perpendicular line and median in a given figure.

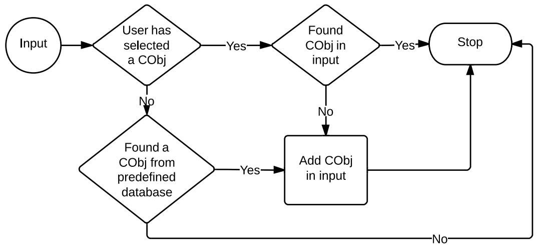

Figure 4: Explains the algorithm for adding construction objects in a given figure.

4.3 Generating Facts and Solutions (GFS)

This component is responsible for finding the values of unknown variables of the generated configuration from the other components. This component acts as question generator and solver. The unknown variables whose values have been found represent the generated questions. The steps that leads to finding the unknown variables represent the solution. The algorithm will consider only those facts which do not include CObj but make use of CObj for reasoning. Figure 3d shows an example of such a fact. The new fact is related to the length of side CB of a triangle. However, it requires the construction of a perpendicular line from E to AB.

There may be several ways for finding the values of the unknown variables. In such cases, this component shows all solutions. For generating the new facts, it uses a predefined database of theorems. The theorems are represented in the form of axioms. We have used The School Mathematics Study Group (SMSG) axiomatic system, which is a combination of Birkhoffs and Hilberts axiom (Francis,2002) systems. More details can be found in an algorithm (annonymous, 2013).

4.4 Generating New Configurations (GD)

This component starts with a search for user-selected theorems. If the user has not chosen any theorem, a theorem is non-deterministically selected from the predefined knowledge database of theorems. Our algorithm terminates when all theorems have been selected once for a given figure configuration. After theorem selection, pattern matching on the theorem figure is performed. If a matching pattern is found, a set of input and/or output values are assigned to the chosen pattern. This assignment is done with the help of a predefined set of theorems. In case no pattern is found, a new theorem is chosen from predefined theorem database, KT, and the whole process is repeated until we get the desired configuration. More details can be found here (annonymous, 2013).

5 IMPLEMENTATION

Each component of our tool is implemented independently, using state-of-the-art libraries and systems.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

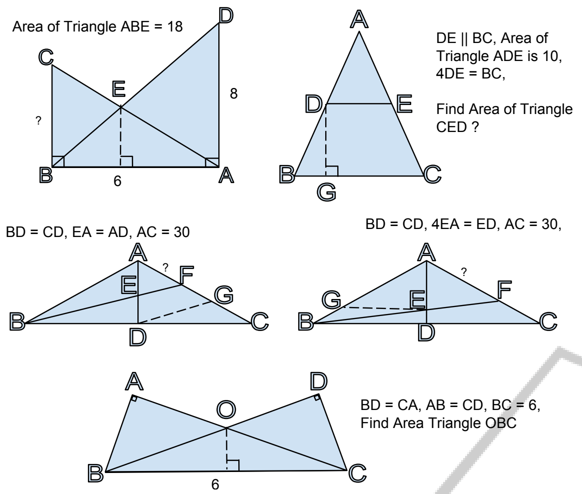

Figure 5: Figure (a) can be solved by drawing a perpendicular from E to AB (b) A perpendicular is drawn from D to CB (c) A parallel line is drawn from D to AC (d) A parallel line is drawn from E to BD (e) Three lines are drawn joining existing points, AR, BP, CQ.

C++ is used for performing calculations and Python is used for implementation of the algorithm and predefined knowledge databases. Constraint Handling Rules (CHR) (Frühwirth and Raiser, 2011) are used for generating new facts from the axioms and the given facts. For implementing knowledge representation, the graph database Neo4j (Vicknair and Macias, 2010) is used. Both questions and a set of predefined theorems are represented using Neo4j.

Results

Our system can generate geometry questions using the framework described in Section 4. Currently, our knowledge database of objects contains line segments and triangles. In addition, we have a predefined set of more than 100 theorems. The generated questions cover four construction objects, e.g. perpendicular line. Figure 5 shows various questions generated by our system on selecting "*triangle*" as object and various CObj such as parallel line, perpendicular line and median. The dashed lines in the Figure 5 represent construction of the CObj required to solve the question.

GRAMY (Matsuda and Vanlehn, 2004) is a geometric system that can solve geometric questions which require implicit construction. However, due to inavailability of GRAMY's source code, we cannot validate the questions generated by our framework.

6 CONCLUSION

In this paper, we provide a framework for the automatic generation and solving of geometric questions which require implicit construction of geometric objects for their solution. Our system is able to quickly generate large numbers of questions on specific topics based on construction objects. Such a system will help teachers reduce the time and effort spent on the tedious and error-prone task of generating construction-based questions.

Future work involves conducting experiments in which the teachers would be asked to differentiate between the textbook questions and the framework generated questions. Another major improvement would be to generate questions according to the required difficulty level in addition to the implicit construction of objects.

SCIENCE AND TECHNOLOGY PUBLICATIONS

REFERENCES

annonymous (2013). Automated generation of geometry questions for high school mathematics. In *Sixth International Conference on Computer Supported Education*.

Chou, S. C., Gao, X. S., and Zhang, J. Z. (1994). Machine proofs in geometry: Automated production of readable proofs for geometry theorems.

Frühwirth, T. and Raiser, F., editors (March 2011). *Constraint Handling Rules: Compilation, Execution, and Analysis*.

Gao, X.-S. and Lin, Q. (2004). Mmp/geometer a software package for automated geometric reasoning. In Winkler, F., editor, *Automated Deduction in Geometry*, volume 2930 of *Lecture Notes in Computer Science*, pages 44–66. Springer Berlin Heidelberg.

Greeno, G., J., Magone, and Maria E. Chaiklin, S. (1979). Theory of constructions and set in problem solving. Technical report, Pittsburg Univ Pa Learning Research and Development Center.

Matsuda, N. and Vanlehn, K. (2004). Gramy: A geometry theorem prover capable of construction. *J. Autom. Reason.*, 32(1):3–33.

R. Singh, S. G. and Rajamani, S. (2012). Automatically generating algebra problems. In *AAAI*.

Schrijvers, T. and Demoen, B. (2004). The k.u.leuven chr system: Implementation and application.

Vicknair, C. and Macias, M. (2010). *A Comparison of a Graph Database and a Relational Database*.

Wilson, S. and Fleuriot, J. D. (2006). Geometry explorer: A tool for generating diagrammatic full-angle method proofs. Technical report, University of Edinburgh.