Automated generation of region based geometric questions

Rahul Singhal
School of Computing
National University of Singapore, NUS
Singapore
Email: rahulsinghal@nus.edu.sg

Martin Henz
School of Computing
National University of Singapore, NUS
Singapore
Email: henz@comp.nus.edu.sg

***Abstract*—We extend our previously proposed framework that combines a combinatorial approach, pattern matching and automated deduction to generate geometry questions which, directly or indirectly, require finding the congruent regions formed by the intersection of geometric objects. The extension involves proposing a knowledge representation for regions and a rule-based algorithm for generation of region-based knowledge representation. In addition, several algorithms such as circle/arc projection to straight line(s) are proposed to avoid numerical reasoning for proving congruent regions, making the solution eligible for high school geometry domain. Furthermore, we propose the integration of this framework with our previously proposed framework to generate questions involving both implicit construction and congruent regions. The system is able to generate the solution(s) of the questions for their validation. Such a system would help teachers to quickly generate large numbers of questions based on several properties of geometric objects such as length, angle, area and perimeter. Students can explore, revise and master specific topics covered in classes and textbooks based on generated questions. This system may also help standardize tests such as Primary School Leaving Exam (PSLE), GMAT and SAT. Our methodology uses (i) a combinatorial approach for generating geometric figures (ii) Pattern matching and rule-based approach for region generation (iii) automated deduction for checking equality of properties of geometric objects (iv) linear equation solver to generate new questions and solutions. By combining these methods, we are able to generate questions involving finding or proving congruence relationships between the regions generated by the geometric objects based on a various specifications such as objects and concepts. Experimental results show that a large number of questions can be generated in a short time. A survey shows that the generated questions and the solutions are useful and fulfills the high school criteria.**

***Keywords*-deductive reasoning; rule-based algorithms; pattern matching; high school geometry**

I. Introduction

Geometric question which involves finding the congruence relationships based on various properties of the geometric objects is one of the major categories of geometric topics covered in high school geometry domain. Students need to have the skills of finding the congruent shapes in the given figure and use these observations to find various properties such as length, angle, area and perimeter involved in the figure. These questions do not follow any set pattern. One can only learn the axioms and results proven from these axioms, and apply them appropriately. Therefore, a question may have (possibly infinitely) many solutions. To practice the necessary problem solving skills, students require a large number of different types of geometry questions on various concepts. However, textbooks and online sites provide a limited predefined number of questions for each topic. In addition, once practiced, these questions lose their purpose of enhancing student thinking. Furthermore, 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 the users, the question's generation framework has applications to other research areas, such as Intelligent tutor systems (ITS) [1] and Massive Online Open Courses (MOOC) [2]. One of the major concerns in MOOC involves dealing with the multiple login of a student with different login credentials to get more marks. This system can help by providing different questions each time the user logs in. In addition, the framework can be useful for various online sites such as Khan Academy [3], which provides large number of questions for practice.

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 question generation based on the regions generated by the intersection of geometric objects. 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. In addition, the systems cannot solve the region based questions due to lack of knowledge representation for processing regions.

The aim of this paper is to develop a framework that can be used to generate the region-based geometry questions based on specific properties, such as length, angle, area and perimeter of geometric objects to be involved in their solution. The solutions will involve finding congruent regions based on user desired theorems and objects. Hence the framework can quickly generate large number of questions to test the several concepts such as length, angle, area and perimeter on various geometry objects.

The main contributions of this paper are as follows:

1) We extend our previously proposed framework to generate geometry questions which were not possible to generate.
2) We introduce a knowledge representation for regions generated by the intersection of the objects.
3) We propose a rule-based algorithm for generating region-based knowledge representation.
4) We provide a substantial evaluation that demonstrates the effectiveness of our generator. We are able to generate questions covered in the textbooks and questions asked in PSLE, SAT and GMAT.

II. Related Work

In this section, we provide a general review of related works. Computational research in the geometry domain started in the 19th century. However, a lack of question generating in geometry in the literature required a principled ab-initio approach in our work.

There are several automated theorem proving systems in the geometric domain such as Geometry Explorer and JGEX [4] that allows users to build proofs based on geometric constructions. The Angle method, Wu's method and Grobner basis method [5] are some of the methods used by the existing geometry systems for automatically proving geometric theorems. Another research [6] proposed a rule-based algorithm for automated exercise generation in Euclidean geometry. However, these methods are either not suitable for the question generation or the approach used is not suitable for the high school geometric domain.

Recently, a template-based algorithm [7] is proposed for automatically exercise generation specially for the embedded system. The algorithm generates a template and requires a question solver to test the validity and the effectiveness of the generated template. However, such approach is not effective in the geometry domain where a question cannot be considered invalid if it not solved by the solver. The reason for the inability may be a lack of predefined theorems to solve the question.

An algorithm developed by Rohit [8] used a numerical approach for solving the geometric questions. The algorithm uses the numerical reasoning approach for finding the solutions. Hence, the solutions generated by this algorithm are out of the scope of the high school mathematics.

A grammar called Shape Grammar is used for generating spatial 2D and 3D shapes. Krishnamurti and Stouffs [9], [10] describe a grammar for performing trivial operations on shapes. The most common operations on shapes are transformation, rotation and scaling. Shape grammars have been studied for their application in computer-aided architecture design and planning and are not directly applicable to question generation; however they served as an important inspiration to our work.

Our recently proposed framework can generate questions based on concepts, theorems and user-selected objects [11].

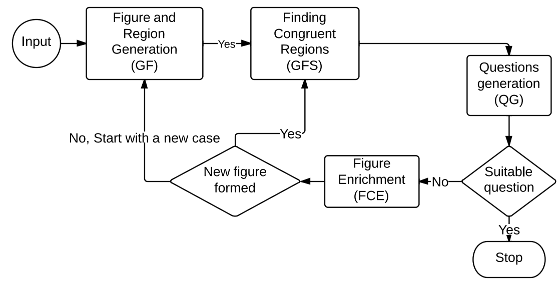

Figure 1. Connection of the components with knowledge databases

In addition, the framework can generate questions which require implicit construction for their solution [**?**]. However, the framework cannot handle the relationships based on the regions formed by the intersection of objects due to a lack of knowledge representation for the regions. Hence, the framework cannot generate questions based on various properties of regions such as length, angle, area and perimeter. The current work is an extension of this framework in order to address the limitation.

III. Specification of the Input for Question Generation Task

Mathematically, the input for generating geometry question by our system can be represented by a tuple (S,Q) where:

- S is a multiset of shapes such as "line", "triangle", "square", etc.
- Q $\subseteq$ {"area", "perimeter", "length", "angle"}

Thus, in order to generate the geometry questions, the user has to provide a set of geometry shapes S and optionally select a query type Q.

The key insight that underlies this work is the use of the rule-based algorithm and an axiomatic system to generate and find congruence relationships between the regions formed by the intersection of objects. In addition, graph based knowledge representation is used for processing regions. The next section describes the framework used for question generation.

IV. Framework

Our framework comprises four major components along with the knowledge databases used for storing the geometric figures. Figure 1 shows the connection of these components. The input consists of objects, concepts and theorems selected by the user. The input is fed into the first component, *generating figure (GFR)* for generation of figures configurations. Each configuration constitutes a diagrammatic schema (DS) [12] and a set of unknown variables representing the relationship between the objects. As a result, a *region graph* is generated which represents the regions formed by the objects. The configuration is passed to the second component, *generating facts and Solutions*

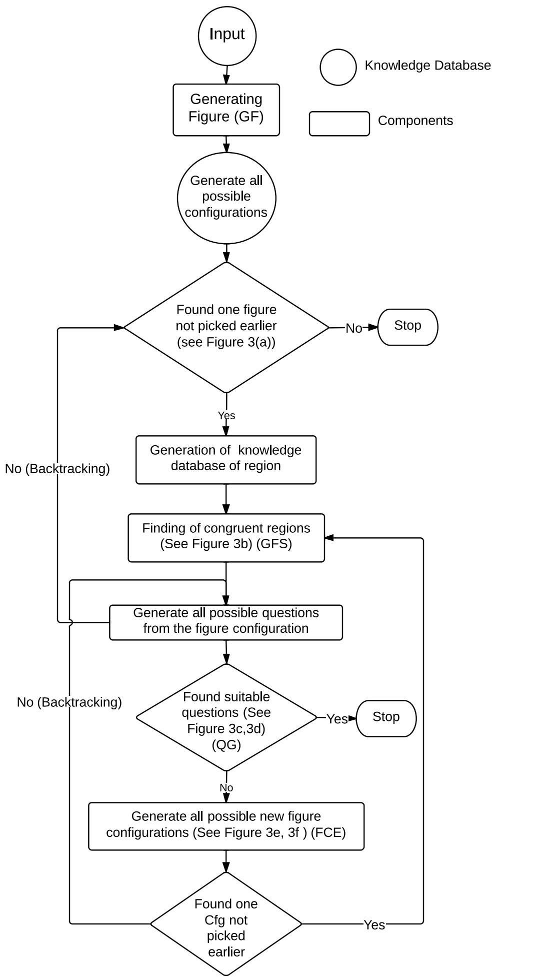

Figure 2. Flow diagram of the algorithm for generating questions

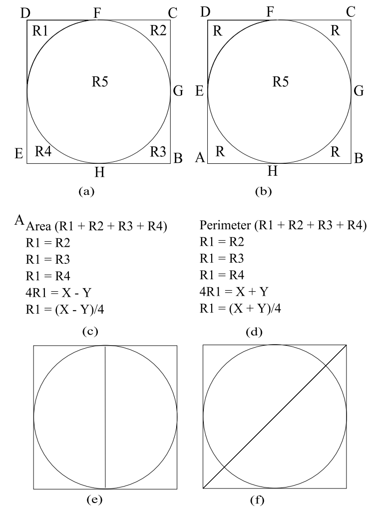

Figure 3. (a) One of the figure generated by using *square* and *circle* as the geometry objects using the GF function. (b) All the congruent regions are detected for (a) using the GFS component. (c) and (d) are the questions and solutions generated by QG component. (e) and (f) are the new configurations generated from figure (a) by FCE component.

(GFS) for finding congruent regions in the given figure. Later, the configuration is passed to the third component, *question generation (QG)* to find values of the unknown variables. The QG generates a set of linear equations from the *Region Graph* data structure. The linear equations are then fed to a semantic equation solver and values of all the variables are calculated and are treated as 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. If the suitability conditions for the generated configuration (Cfg) are not met then the configuration is fed into the last component, *figure configuration enrichment (FCE)* to add the information (e.g. a new shape and/or new constraints) to the existing configuration. The new figure configuration is then fed to the second component for the region's graph modification. This process continues until a question meeting the suitability condition is generated.

A. Algorithm

Figure 2 represents the flowchart describing the algorithm for generating questions. Figure 3 shows the step by step execution of the algorithm. Currently, we have restricted the geometric objects to circles and squares only. Hence, the following input is selected in our example.

- S: ”square” , ”circle”
- Q: Area and Perimeter

In the following subsections, we will describe each component and their interaction with the knowledge databases.

B. Generating Figures and Regions (GFR)

GFR is further divided into two sub-components namely, *Figure configuration generation and Region (RG) generation*. The following sub-sections explain the two components in detail.

1) Generating figures from the user input (GF): This is the first step executed by the algorithm described in Figure 2. This component generates figure configurations through the various predefined combinations of objects. Currently, only circles and squares are supported. Hence the algorithm includes all combinations in which circles and squares can intersect. Our algorithm is limited to the intersection of maximum of four circles and squares. Adding more objects can make the figure more complex, possibly leading to unusual questions that are not helpful for practice.

2) Generation of the Region Graph (RG): The RG generation component generates a data structure, RG, representing the regions in the given figure. A region may be described as a closed shape bounded by curves and each curve may be a straight line or a circle/arcs segment. The intermediate

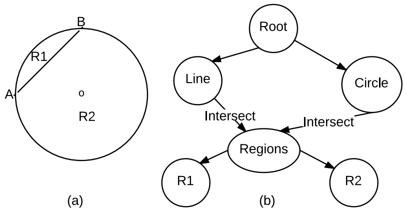

Figure 4. Figure explaining formation of two regions R1 and R2 from an intersection of line and a circle.

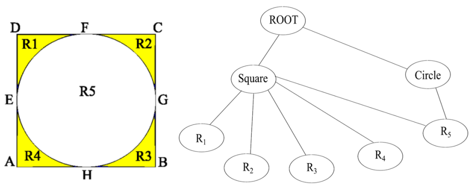

Figure 5. A figure configuration and its abstract RG diagram generation from the axiomatic system.

nodes of the region graph consist of the objects whose intersection results in the formation of the regions. The end nodes represents the generated regions. The links joining the nodes shows the method of combination of geometry objects such as touch and intersect. Figure 4a shows the two regions formed by the intersection of a line and a circle and Figure 4b represents the generated RG from it.

One important property of RG is that the area of the closed shape is equal to the sum of the areas of all regions directly connected to it. Figure 4b shows that the area of circle is equal to the sum of the area of R1 and R2. Figure 5 shows a figure configuration used in Figure 3 and its abstract RG diagram, in which, the area of the sqaure is equal to the sum of the areas of all the regions (R1 to R5).

The Region graphs play a crucial role in our framework as they are passed to other components and questions are generated from them. A Region garph is generated from a given figure configuration generated by the GF component. A rule-based algorithm is proposed for the RG generation, where, each rule refers to a particular configuration and add information about the newly generated regions to the RG. A Pattern matching concept is used to find the exact rule and corresponding modifications are done to the RG graph. Forward chaining based on axiomatic approach is used for RG generation where each rule is represented in form of an axiom. For example, a Prolog query 1 refers to the axiom which satisfies Figure 4a and Figure 4b represents the generated RG from it. Similarly, Figure 5 shows a figure configuration used in Figure 3 and its abstract RG diagram.

$$\begin{aligned} lineIntCircle(a,b,o,r,l,r1,r2) :- & incident(a, circle(o,r)), \\ & incident(b, circle(o,r)), \\ & line(a,b), \\ & intersect(line(l), circle(o,r)) \end{aligned} \tag{1}$$

We have used The School Mathematics Study Group (SMSG) axiomatic system, which is a combination of Birkhoff's and Hilbert's axiom (Francis,2002) system. Next sub-component describes the algorithm for finding congruent relationships in the generated regions.

Data: Region graph (RG)
Result: Modified RG graph representing the congruent regions in the given figure configuration

1) Convert all circles/arcs to the straight lines using Algorithm 2.
2) Apply axiomatic system to the generated configuration.
3) Pick all the combinations of n regions - $\binom{n}{2}$.
4) For each combination of regions, call equality checking algorithm (Algorithm 3).

Algorithm 1: Algorithm for finding congruent regions in a given figure configuration

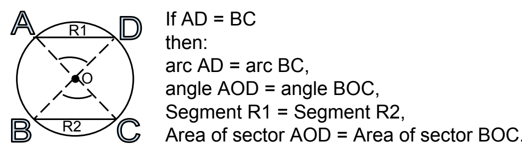

Figure 6. Figure explaining the area equality of two congruent chords of a circle

C. Generating facts and solutions from the figure configuration (GFS)

GFS is responsible for finding the hidden relationships in a given configuration. The new hidden relationships, not possible in previous framework, refer to finding the congruent regions. This component introduces several algorithms to avoid numerical reasoning for proving the congruent regions making the solution suitable for high school geometry domain. The following sub-sections explain the algorithms in detail.

1) Finding the congruent regions: The congruent regions refer to regions which are equal in area and/or perimeter. Selection of area and/or perimeter for congruency, depends on the user-selected option. Algorithm 1 is used for finding the congruent regions in a given configuration and uses various algorithms internally such as projection of arcs to straight lines and *Equality checking algorithm*(see next subsection) for finding the congruent regions in a given figure and add this information to the existing RG data structure.

Algorithm for circle/arc projection to straight line(s): The motivation behind this projection is the generation of non-computational reasonings for the facts regarding area and perimeter. The conversion of circle/arc to a straight lines is done using *Congruent Chords Conjectures* [13].

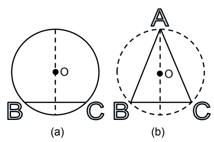

Figure 7. (a) Figure configuration where an arc does not cover the whole circle according to the algorithm (b) Projection of major arc of the circle to two straight lines

Data: Region graph (RG)
Result: A modified region graph with added lines corresponding to the existing arcs

1) Get all the chords from the intersection of other objects on the given circle.
2) Check if the whole circle/arc is covered through the chords generated in step 1 with the help of two rules.
   a) The sum of the angles subtended by all chords at the center of the circle should be 360^o.
   b) Each angle should be less than or equal to 90^o.
3) If the two conditions mentioned earlier are met, then terminate. Otherwise, goto next step.
4) A chord which does not follows conditions mentioned above, a new point and two new lines are introduced. New point will be the intersection of perpendicular bisector of the chord with the circle. The two lines are generated by joining new point with the end points of a given chord. Figure 7 shows an example of this conversion. Line segment AB and AC represents the area covered by arc AB and arc AC respectively.

Algorithm 2: Algorithm for projecting circle on a line

Following the conjectures, Figure 6 shows that the congruence of straight lines AB and CD can be used for checking of area equality of the regions R1 and R2. Algorithm 2 describes the method for projection of a circle to straight line(s) for area and perimeter comparison based on the conjecture. Figure 8 shows an example of projection of a circle to the straight lines.

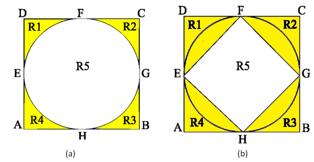

Figure 8. Figure explaining finding congruent region for curve regions.

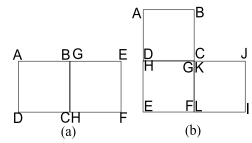

Figure 9. (a) Two squares sharing common side through BC and GH (b) Two squares are indirectly sharing sides through DC and KL.

The axiomatic system is used for finding the congruent regions from a given figure configuration. Firstly, the GFS axiomatic system [11] is applied to find the hidden relationships in a given figure configuration through forward chaining. Several cases arise for checking the equality of sides and angles such as sharing common side and indirectly sharing equal sides. Figure 9 shows example of each case. In order to cover such cases, a different set of axioms is used. The following is a simple example for the axioms used :

Axiom 1. *All sides of a square are equal*

Equality checking algorithm: This subsection describes the algorithm for finding the equality among the two regions. Algorithm 3 describes the algorithm for finding the congruent regions in the figure. Figure 10 shows the two regions which will used as an input.

Now RG contains all the information regarding congruent regions. Next section deals with the generation of questions from the RG structure.

D. Questions Generation from the Region Graph (QG)

This section deals with the algorithm for question generation. Algorithm 4 describes the steps for question generation from the RG. A RG is divided into various levels, where, Level 0 contains root node and the objects at Level 1 are the user-selected objects, for which we will have information about the area and perimeter. Nodes at Level 2 are compo-

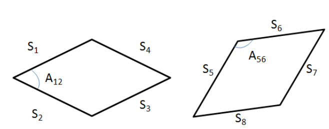

Figure 10. Figure showing two regions for checking of congruency

Data: Two regions R1 and R2, with sides and angles (Figure 10)
Result: Modified RG

1) Check the number of bounded regions (should be same). If not same then terminate.
2) Check the order and type of bounded curves (*Order* refers to the clockwise ordering of curves should be same in both regions and *Type* refers to the degree of the curves should be equal in both the regions). If not same then terminate.
3) Start with one side of region R1 (say S1), check with the matching side in R2.
4) If a matching side is not found, terminate. Otherwise, save all the matching sides.
5) For each matching side, pick an adjacent side of R1 and check with the adjacent side of the selected side of R2.
6) If not found in all the saving sides, terminate.
7) If found one side, check for the equality of the angles between the two selected sides of each region. This step is optional in case of finding regions of equal *"perimeters"*.
8) If found one side, goto step 5 and repeat the above process till all sides of region R1 match with the distinct sides of region R2.
9) If both regions are equal, add this information to RG and terminate.

Algorithm 3: Algorithm for finding equal regions in a given figure configuration

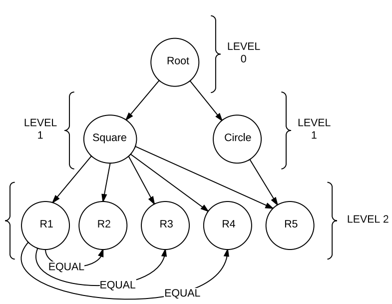

Figure 11. Figure explaining levels in region graph

nents of the nodes at Level 1. Figure 11 shows the level's concept used by the Algorithm 4. There are three levels shown in Figure 11. The arrows pointing from Level 1 to Level 2 indicate the combination of regions. In Figure 11, the sum of regions R1 to R5 will be equal to the area of the square.

Figures 3c and 3d show some of the questions and the

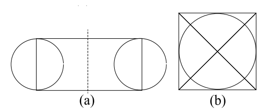

Figure 12. Figure explaining generating new configurations from an existing configuration via (a) adding mirror image (b) via adding line of symmetry

solutions generated for Figure 3a based on algorithm 4.

Data: Region graph (RG)
Result: Question(s) with solutions.

1) Generate all linear equations from nodes at each level.
2) Generate other equations from several relations such as *equality, addition and subtraction*.
3) The equations are then feed into a semantic equation solver and the questions are generated along with the solutions.

Algorithm 4: QG: Algorithm for generating questions from a given RG

E. Generating new configurations from an existing configuration (FCE)

The FCE component generates a new configuration from an existing configuration. The various ways of obtaining new configuration are as follows

1) Adding mirror image: This method will add symmetry in the existing configuration. Mirror image can be obtained by performing the similar steps done earlier to obtain the existing figure configuration. However, we need to add the objects in the opposite directions with respect to existing figure configuration. Figure 12a shows an example of adding the mirror image of a circle.

2) Adding a line of symmetry: The motivation behind this method is to generate a symmetry in the existing configuration by adding a line of symmetry. A line is a line of symmetry for the entire configuration if it is a line of symmetry of each object. Figure 12b shows an example of adding a line of symmetry to Figure 3.

V. Generating question involving implicit construction

Our proposed algorithm [**?**] can generate geometry questions requiring implicit construction for its solution. The algorithm works on the concept of adding *hidden* objects in the GFS component. The algorithm proposed in section 2 can be integrated with this algorithm. The major change would be the addition of constraints in GFS component resulting in the generation of those facts which contain

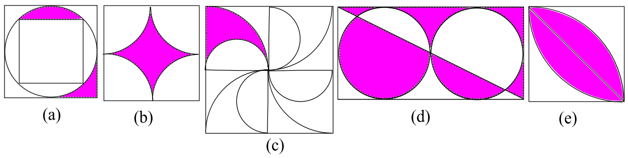

Figure 13. Generated questions based on "circle" and "square" as inputs. The generated question in each figure is finding the area and perimeter of the shaded region. (a) describes a circle inscribed and circumscribed a square (b) covers quadrant of a circle and a square (c) covers semicicrcles, quadrant of a circle and a square (d) generates adjacent circles and squares (e) covers semicircles and square

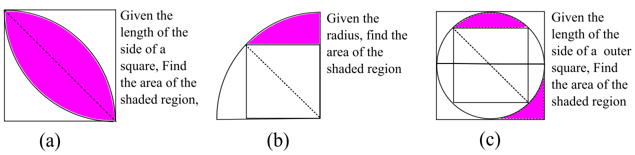

Figure 14. Generated questions based on "circle" and "square" as inputs. The generated question in each figure is finding the area and perimeter of the shaded region. (a) and (b) covers semicircles and square (c) describes a circle inscribed and circumscribed a square

the construction object and the reasoning behind the new fact does not involves the construction object. However, the reasoning must be based on relationships involving congruent regions. Figure 14 shows various examples in which dotted line represents the implcitly drawn construction object.

VI. Implementation

Each component of our tool is implemented independently, using state-of-the-art libraries and systems. The programming language Python is used for implementation of the GF and FCE component. The algorithm in GFS is implemented using Constraint Handling Rules (CHR) [14]. We use the CHR library provided by K.U.Leuven, on top of SWI-Prolog [15]. For implementing the knowledge representation RG, the graph database Neo4j [16] is used.

Experimental Results

We are able to generate geometry questions using the framework described in the *Framework* section. Currently, our knowledge database of objects contains circles and squares and a predefined set of more than 100 theorems. Figure 13 shows various questions generated by the system on selecting "*square and circle*". Questions in Figure 13 are given a length of a side of a square and the questions would be finding the area and perimeter of the shaded regions. The generated questions implicitly involve finding congruent regions. Figure 14 shows the generated questions which involve both implicit construction and implicit finding of congruent regions. Dotted line shown in Figure 14 represents the implicit construction needed for solving the question.

In addition, we did a survey containing the mixed questions, some question are from our generated system and some are from textbooks and online sites. The teachers are asked to classify the questions in the above mentioned categories. In addition, the teachers are asked to give the comments on the solutions and the usefulness of the questions.

Evaluation

A pilot evaluation was conducted in order to estimate the feasibility of the whole approach and generated questions. We entered 10 rules in the system, which correspond to specific geometric theorems: Similarity in triangles, Pythagorean Theorem, Basic Trigonometry formulas.

Different type of inputs are given to generate various different problems. Object includes circles and squares. The Concept includes, the properties of square and circle along with the geometry theorems useful for finding the congruent regions. In addition, options of generating questions which involve implicit construction along with finding congruent regions are given. Some of the questions have multiple ways of finding the relationships. The prototype generated a large number of problems, some of which were isomorphic, that is, identical from a pedagogical point of view. Note that the problem illustrated in Fig. 1 was among the ones generated. All problems were correct, as checked by the author of this paper. From these, 10 problems were selected for evaluation, covering almost evenly the range of all calculated values in interest and diversity. The small number of problems was due to evaluator's limited time availability.

The real users for our survey are high school teachers, students and professionals involved in standardized exams like GMAT and SAT. Selected problems were given to several experienced High School Mathematics teachers from different parts of the world, for ex., India, Singapore and US. In addition, hundreds of students, both high school and college level, were involved in the survey.

For each problem, the assumptions, known relationships and few unknown relationships to be proved/ found out as well as automated generated diagram were given to the participants. 6 survey questions are asked for each geometric question, such as the quality and the appropriateness of the generated solutions. Table 1 shows the questions asked and the statistics of users response. It can be seen that various teachers have different perception regarding the quality and difficulty of the question to be given in exam.

The table shows that half of the users considers these questions as new one. However, this depends on the domain knowledge and memory. There are few questions which are considered as new by almost each user. In addition, most of the users are in consensus with the usefulness of the generated questions. Furthermore, most of the users have considered the generated solution of the questions appropriate for high school mathematics. The options for providing

Users	Has seen Online/Textbook (percentage)	Appropriateness of solution (percentage)	Input used (percentage)	Concept used (percentage)	Quality (percentage)
High school teachers	10	80	100	90	90
College level teachers	60	80	95	95	90
High school students	10	60	100	95	95
College level students	5	70	100	90	95

Table I
SHOWS CONFIGURATION GENERATION IN DIFFERENT DOMAINS FROM INPUT

objects and concepts for generating these questions matches with the objects and concepts actually provided in the system. Overall, it can be seen that the system is able to fulfill the aim of generating questions quickly from the given input along with the appropriate answer.

Although the number of participants is very limited and the report of the above data is anecdotal, we see that for both diversity and quality for testing, the concordance between teachers. Although these results by no means can be generalized, they are hopeful initial indicators of the potential validity of the proposed measures for exercise selection, proving a useful basis for further justification and/ or adjustment.

VII. CONCLUSION

In this paper, we provide a novel framework for the automatic generation and solving of questions based on regions and their properties such as length, angle, area and perimeter. Such a system will help teachers reduce the time and effort spent on the tedious and error-prone task of generating questions and will support new developments in education such as ITS and MOOC. Our framework aims to develop an automated geometry question generation system that uses an axiomatic approach for finding the congruent regions and uses a semantic linear equation solver to generate questions and solutions. Both the questions and their solutions can then be translated into other natural languages.

Future work can be carried in various directions. One major improvement would be using the user-feedback to quickly generate user-desired questions. Furthermore, generating questions according to the required difficulty level is one of the hard challenge. Other improvement would be automated addition of knowledge by the user.

REFERENCES

[1] J. Shute and J. Psotka, "Intelligent tutoring systems: Past, present, and future. human resources directorate manpower and personnel research division." 1994.

[2] L. Yuan and S. Powell, "Moocs and open education: Implications for higher education," in *Publications from the Centre for Educational Technology, Interoperability and Standards (cetis)*, 2013.

[3] Khan, "Khan academy." [Online]. Available: https://www.khanacademy.org

[4] X.-S. Gao and Q. Lin, "Mmp/geometer asoftware package for automated geometric reasoning," in *Automated Deduction in Geometry*, ser. Lecture Notes in Computer Science, F. Winkler, Ed. Springer Berlin Heidelberg, 2004, vol. 2930, pp. 44–66.

[5] S. C. Chou, X. S. Gao, and J. Z. Zhang, "Machine proofs in geometry: Automated production of readable proofs for geometry theorems," 1994.

[6] A. Papasalouros, "Automatic exercise generation in euclidean geometry," in *In Proceedings of Artificial Intelligence Applications and Innovations*, 2013.

[7] D. Sadigh, S. Seshia, and M. Gupta, "Automating exercise generation: A step towards meeting the mooc challenge for embedded systems," in *In Proceedings of the Workshop on Embedded Systems Education (WESE).*, 2012.

[8] R. Singh, S. Gulwani, and S. Rajamani, "Automatically generating algebra problems," in *AAAI*, 2012.

[9] R. Krishnamurti and R. Stouffs, "Spatial grammars: motivation, comparison, and new results," in *Proceedings of the fifth international conference on Computer-aided architectural design futures*, ser. CAAD Futures '93. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1993, pp. 57–74.

[10] R. Stouffs, "The algebra of shapes," Ph.D. dissertation, TU Delft, 1994.

[11] R. Singhal, M. Henz, and K. McGee, "Automated generation of geometry questions for high school mathematics," in *Sixth International Conference on Computer Supported Education. Accepted for publication as full paper*, 2013.

[12] Greeno, G. James, Magone, E. Maria, Chaiklin, and Seth, "Theory of constructions and set in problem solving." Pittsburg Univ Pa Learning Research and Development Center, Tech. Rep., 1979.

[13] dwiggins, "Congruent chords conjecture," 1996. [Online]. Available: http://www.geom.uiuc.edu/ dwiggins/conj38.html

[14] T. Frühwirth and F. Raiser, Eds., *Constraint Handling Rules: Compilation, Execution, and Analysis*, March 2011.

[15] T. Schrijvers and B. Demoen, "The k.u.leuven chr system: Implementation and application," 2004.

[16] C. Vicknair and M. Macias, *A Comparison of a Graph Database and a Relational Database*, 2010. [Online]. Available: http://www.neo4j.org/