

GPT: A Tutor for Geometry Proving

Roscoe Nealle ALICBUSAN^{a*}, Laurence Nicholas FOZ^a, Paolo Vittorio MERLE^a,
Ethel Chua Joy ONG^a & Minie Rose LAPINID^b

^aCollege of Computer Studies, De La Salle University, Manila, Philippines

^bBr. Andrew Gonzalez FSC College of Education, De La Salle University, Manila, Philippines

*roscoe_nealle_alicbusan@dlsu.edu.ph

Abstract: Geometry is a mandatory subject for secondary school students, where they learn geometric figures and their properties, and the axioms, postulates and theorems involving them. A key topic in Geometry class is proving, where learners are required to derive and prove a certain property is true based from the given properties and by using various axioms, postulates and theorems. This is where most learners encounter difficulty. In this paper, we describe Geometry Proof Tutor, a learning environment where learners can practice Geometry proving through multiple representations - two-column proof and proof tree. With the use of a knowledge base to model Geometry concepts, the software validates the learner's proof statements and provides corrective feedback accordingly. Test results showed that the learners found the availability of the proof tree to be useful in tracking their progress. The presence of complete reasons to use for proofs also helped them understand Geometric Proving better. However, the long list of available choices and the one-proof-statement-at-a-time user interface design made it difficult for them to encode their answers.

Keywords: Geometry, Geometry Proving, Knowledge Representation, Learning Environment

1. Introduction

Geometry is a branch of mathematics that has been around since the time of the Egyptians, when the Great Pyramids were built. It defines and relates the basic properties and measurement of line segments and angles, and Euclidean Geometry (Gantert, 2008).

Geometry is a subject taken by all students in secondary schools. Students learn geometric figures, their various properties and the axioms, postulates and theorems involving them. These lessons build upon the basics taught in elementary school - from recognizing a triangle to knowing that the sum of its angles is 180 degrees (Triangle Angle-Sum Theorem). The next and usually final step in their lessons involve proving, wherein students are required to derive and prove a certain property is true based from the given properties and the various postulates, axioms and theorems discussed in class.

Several models and frameworks have been developed to teach Geometry. The van Hiele model of Geometric Thought, which was proposed at the University of Utrecht back in 1984, posits that there are different levels of geometric understanding which the learner must go through sequentially from simplest to the most complex level (Crowley, 1987). These five levels, arranged from simplest to the most complex, are: Visualization, Analysis, Informal Deduction, Formal Deduction, and Rigor.

Visualization encompasses the most basic understanding of geometry. It is in this level that learners recognize shapes shown to them; however, properties and characteristics of the figure are not yet evident. At the Analysis level, learners begin to examine and observe figures and would be able to determine characteristics that define them. In the Informal Deduction, learners begin to understand how characteristics and figures relate to each other. Formal Deduction, on the other hand, is the level wherein learners have the capability to see the relationship of many different factors, terms defined and undefined, as well as concepts they may or may not know, and are able to create their proofs in many ways. Finally, learners in the Rigor level can work independently from the Euclidean system and they see geometry in abstraction (Crowley 1987).

DeFT (Ainsworth, 2006) is a conceptual framework designed to make representational systems easier to understand and to maximize the benefits from multiple external representations (MERs) without too much cognitive cost for the learner. This framework proposes that the effectiveness of

MERs depends on three factors: the design factors unique to learning with MERs, the functions they serve to facilitate learning, and the cognitive tasks students must undergo when interacting with MERs.

In a Geometry class, the teacher discusses the topic with the use of diagrams. While each proof is written and discussed, learners jot down notes and simultaneously try to comprehend the lesson (Ainsworth, 2006). The teacher may then opt to give tests to assess the level of understanding of each learner, and subsequently adapts his/her approach accordingly. But checking the learner's work requires a significant amount of time, with the teacher analyzing the students' proofs to determine their correctness and misconceptions. This reduces the amount of time available for the teacher to give individualized feedback.

Research in intelligent tutoring systems and learning environments have designed their tutors to be adaptive to the needs of the individual learner. These tutors have been equipped with the necessary knowledge to analyze student misconceptions, to provide corrective feedback, and to devise teaching strategies to match the individual needs of the learners. In this paper, we present Geometry Proof Tutor (GPT), an intelligent learning environment for practicing Geometry proving. We discuss its strategies in analyzing errors in student proofs and providing corrective feedback.

2. Related Work

Geometry Visualization Software: Software has been developed to provide learners with visualizations of geometric figures or proofs. The study of Baccaglioni-Frank et al. (2013) has shown positive benefits in using software to visualize geometric concepts, especially those used in proving. Geometer's Sketchpad (<http://www.dynamicgeometry.com>) is a commercial software that provides its user tools to draw and create geometric shapes, objects and functions. It also allows users to alter and manipulate these figures through a simple user interface. GeoGebra (<https://www.geogebra.org>) is a similar free software which allows users to draw Geometric figures and also supports additional functions for Mathematics outside of Geometry at the cost of having more complex user interface. Cinderella (<https://www.cinderella.de>) is another software capable of visualizing geometric elements as well as other branches of Mathematics such as Calculus and Discrete Mathematics.

Geometry Proving Software: Tutors were also developed to facilitate learning geometric proving. Matsuda and VanLehn (2005) developed Advanced Geometry Tutor (AGT) to test two geometry proving strategies – forward chaining (proving starts at the given propositions and postulates are applied until it reaches the goal) and backward chaining (proving starts at the goal and postulates are applied in reverse until the given is reached). GeoProof (<http://home.hna.org/geoproof/>) is a free software which is vastly different from AGT. It does not offer the two-column proof view that most software have. Instead, it allows users to input a hypothesis regarding the given geometric figure and a conclusion; it then determines if this is true or false.

Multiple Problem Representations: Multiple representations are proposed by the DeFT framework as a means of assisting learning. MR Geo, a software by Wong et al. (2011), was developed to determine if displaying the geometric problem using different visualizations had any effect to students' learning ability with regards to geometric proving. The software provided three different means of viewing the geometry problem: (1) word description, (2) static image, and (3) dynamic image. It also provided two ways of viewing the solution: (1) formal proof and (2) proof tree.

Feedback System: Feedback systems are essential for tutoring systems as it is the means for them to help their users in learning. Fluckiger et al. (2010) cites that learning is a cyclical process of continuous self-assessment and therefore, feedback generally facilitates this. Nyquist (2003) classifies this assessment into categories, namely: (1) weaker feedback only, (2) feedback only, (3) weak formative assessment, (4) moderate formative assessment, and (5) strong formative assessment. Previously mentioned software fall into the first category, which only inform users if their inputs are correct or incorrect and nothing further. Similarly, the DOST Courseware Project (<http://courseware.dost.gov.ph>) is a tutoring system designed by the Philippine's Department of Science and Technology for secondary level Science and elementary and secondary level Mathematics. It stops the user from proceeding further upon entering incorrect answers; however, nothing else is mentioned. SalinLahi III, developed by Regalado et al. (2015), is another tutoring software designed to tutor heritage language learners in Filipino grammar and sentence construction. SalinLahi III gives more

comprehensive feedback using template-based NLG by categorizing the user input as (1) correct, (2) incorrect, (3) near the answer, and (4) cannot be understood.

3. The Geometry Proof Tutor

GPT maintains a bank of Geometry proving problem sets that are designed by teachers through the Editor facility. Every proving problem has a set of similar characteristics (Bass et al., 2001). These are the diagram visualizing the geometric figure, the written statement of the given geometric figure along with the statement to be proven, the statement of facts considered to be true, and the reason for validation (why the statement is true).

3.1 Presenting Geometry Proving Problems through Multiple Representations

Geometry proving problem sets are answered through the interface shown in Figure 1. As Duval (1998) pointed out, learning Geometry involves three cognitive tasks: visualization, construction and reasoning. When deriving a Geometry proof, the learner must be able to understand the given problem from the problem statement, and the figure that comes along with it. The learner then uses his/her knowledge in Geometry and proving to derive proof statements that may lead to the conclusion.

There are three types of representations that are useful for geometric proofs (Wong et al., 2011) - the problem representation, the visual representation and the proving representation. As shown in Figure 1, the problem representation (found on the left side of the user interface) is in the form of a text to help the learners understand mathematical symbols and language, and the logical relationship between all given conditions and the end goal condition. The problem representation is derived from the test bank of proving problems as defined by Geometry teachers.

The visual representation complements the problem representation, to allow the learner to infer from one representation the information that may be hidden or lacking in the other. This approach can help strengthen their understanding of the problem as well as help them find clues on how to get to the solution (Wong et al., 2011).

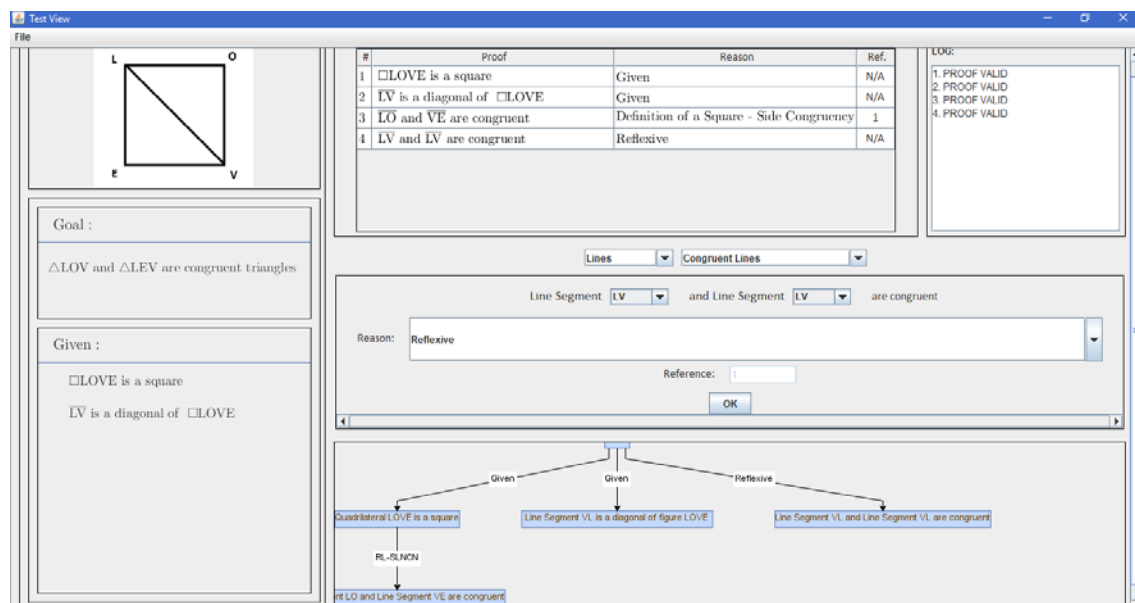


Figure 1. Space for Practicing Geometric Proving

The third representation contains the proof made by the learner. There are two types that can be used: the traditional two-column proof, and the proof tree. Figure 2 shows a sample of the two-column proof, consisting of statements and their corresponding reasons. The two-column proof is shown in the middle part of the user interface (Figure 1). The proof tree is a tree-like structure to represent the network of inferences (Matsuda & VanLehn, 2005). It has the following properties (Wong et al., 2011):

- A root node containing the goal condition to be proven.
- Leaf nodes which are the given facts or self-evident conditions.
- Derived nodes which are statements derived from other derived nodes or the leaf node.
- Edges represent the explicit logical connection between nodes.

When a learner adds a correct proof statement to his/her two-column proof, GPT automatically updates the proof tree, as shown in the lower portion of the user interface in Figure 1.

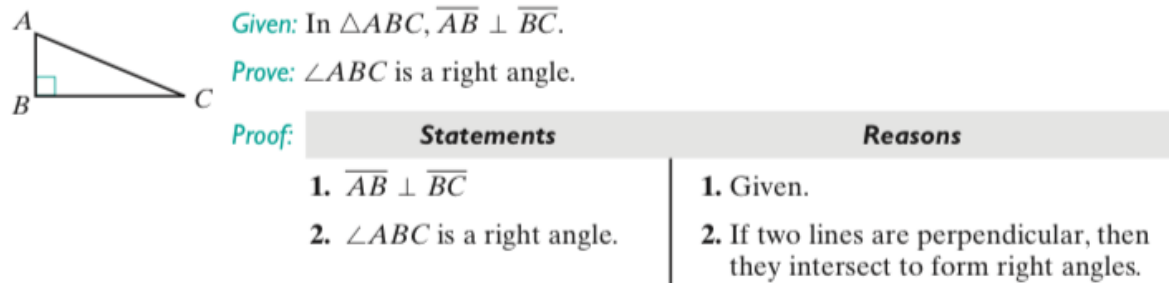


Figure 2. Example of Two-Column Proof

3.2 Proof Checking and Returning Feedback

Whenever the learner inputs a statement to be added to the proof, GPT evaluates the statement through a series of steps. To better illustrate this, the proving problem shown in Figure 2 will be used as a running example.

The first step is a simple preliminary check to look for special cases, such as when the learner identified the statement as part of the given. If so, the GPT checker must cross-check the statement with the list of given. For example, the first statement added to the proof in Figure 2 has “Given” as its rationale. As such, in this step, this statement is cross-checked (*Line AB* is perpendicular to *Line BC*) in the list of given. The special case is vastly different from any other cases and as such, would diverge completely from the other steps. That is, if a statement is found to be a special case, all succeeding steps would be skipped.

The second step involves template checking, which simply verifies the template used by the statement and if there is a rule which results in that statement. For instance, for the second statement in Figure 2, GPT checks if there exists a rule which results in the template “[A1] is a right angle” and would find the “*Right Angle Theorem*” rule.

The third step checks if the stated rationale is correct and there exists a subrule corresponding to it. Subrules correspond to the different rationales under the rule. For instance, under the rule “*Triangle Congruence*”, the subrules include “*Side-Side-Side (SSS) Congruence Postulate*” and “*Side-Angle-Side (SAS) Congruence Postulate*”. For some rules where there only exists one subrule (meaning that there is only one postulate, axiom or theorem that would result in that statement), their names would be similar. In the running example in Figure 2, this would be the case - “*Right Angle Theorem*” rule and subrule.

In Geometry proving, it is common for one proof statement to rely or be related to previous proof statements. Thus, the final step involves checking the referenced statements used in the proof. For instance, the “*Side-Side-Side (SSS) Congruence Postulate*” states that two triangles are congruent if their sides are congruent with one another. As such, entering this rationale into GPT would require three proof statements corresponding to each pair of sides which are congruent. In the running example in Figure 2, the second statement would require a statement stating that the two lines are perpendicular to one another. As such, the previous statement (“*Line AB* is perpendicular to *Line CD*”) would be referenced for the newly added statement to be valid.

If at any point, the GPT checker finds an error in the student’s proof, the checking stops and the incorrect proof statement is highlighted with a corresponding feedback as a form of weak formative assessment. According to Nyquist (2003), weak formative assessment gives feedback coupled with some explanations. As such, the generation of feedback also utilizes templates. The templates used are divided into two categories - templates for correct answers (e.g. “You are correct!”) and templates for

wrong answers (e.g. “The statement [A1] is not reflexive. The reflexive property will only result in statements of congruency.”).

The predefined templates are not commutative. There are instances wherein some of the referenced geometric figures can be interchanged. An example of this is the congruence template, where if *Figure A* is congruent to *Figure B*, then it should follow that interchanging the indices in the template, i.e., *Figure B* is congruent to *Figure A*, would be equivalent. However, the non-commutative design of the templates does not support this. That is, GPT does not have the knowledge that some indices can be interchanged. This can be problematic when learners try to interchange the indices with the idea that they are equivalent. It also introduces problems during checking. For example, in the *SSS Congruence Postulate* checking, if the learner’s input goal is to prove that *Triangle LOV* is congruent with *Triangle LEV*, the checking algorithm will return errors should the learner interchange the congruence statements (i.e. *VE* is congruent with *LO* instead of the other way around).

4. Evaluation, Results, and Discussion

GPT was evaluated by 23 tertiary level students. Each participant used GPT to solve a proving problem. The participant’s proofs and GPT’s responses were internally logged for further analysis. A debriefing interview was conducted to solicit feedback regarding their experience. They also accomplished survey forms to provide a quantitative evaluation of the software.

Table 1 summarizes the average evaluation scores from end-user surveys, using a rating scale of 1 to 5, with 1 representing strongly disagree, 2 for disagree, 3 for neutral, 4 for agree, and 5 for strongly agree. Overall, GPT received an evaluation score of 4.08. The first criterion, “*I can easily add proof statements.*” got the lowest score. This is attributed mainly to the difficulty encountered by the participants in searching for reasons due to the presence of too many choices which are not organized in any manner. Furthermore, proof statements can only be entered one at a time.

Table 1: Evaluation Scores of GPT from the First Iteration of End-User Testing

Criteria	Evaluation Score
I can easily add proof statements.	3.91
The representation of the problem is appropriate.	4.09
The proof tree assists in showing me how my proof has progressed so far.	4.22
The proof tree is easy to read and understand.	4.17
The feedback was helpful in letting me know why my proof was wrong.	4.00

The rest of the criterion received scores above 4.00, with participants stating that they can easily see the possible answers. They found the feedback to be helpful in identifying errors in their proofs. The presence of complete reasons to use for proofs also helped them understand Geometric proving better. Among the criterion, the proof tree received the highest score. It was observed that participants interacted with the proof tree, with some of them moving the view around to see the parts of the tree obscured from their vision. They also used the proof tree as a guide to determine how far their proof has progressed and what their next step would be.

5. Conclusion and Future Work

Geometry Proof Tutor is a learning environment that evaluates the proofs and inferences placed as input by the learners, and gives corrective feedback accordingly. The use of multiple representations allows learners to input their answers using the two-column proof format, and view the corresponding visual

representation generated by the software as a proof tree. An editor facility is also included to allow teachers to define Geometric proving sets.

Test results showed that the participants found the proof tree and the feedback to be useful during a learning session, as these enabled them to track the progress of their proofs and to understand the reasons behind incorrect answers. However, the user interface design needs to be improved to enable a more efficient mechanism for the learners to input their proof statements, including the lookup facility to search for reasons.

Currently, the logs that note down why a learner's proof statement is valid or not (Figure 2), and the retention of the student's mistakes are used to evaluate the knowledge of the learner in geometric proving. However, this only holds for the current problem set. Since GPT is not a full-blown ITS, it does not maintain a student model to track an individual learner's understanding of the different topics in Geometry. Mistakes committed by a student in one session are not stored and used in subsequent sessions. Future research should consider the use of a student model in determining a specific learning goal for each session in GPT. The generation of feedback can also be tailored to the performance and needs of the individual learner.

The algorithm that checks a learner's proof relies heavily on the knowledge representation of Geometry properties, axioms and postulates. As such, it is unable to correctly check a proof if the said proof is not represented in GPT. Examples of these are rules such as conic sections or arcs, which are currently not included in GPT's scope. Other examples include special proving methods outside the traditional proving methods, such as proof by construction - a proving method wherein an additional figure is added to the given figure to give light to more properties.

Although the use of templates to represent Geometry concepts currently has its limitations, amassing learners' usage of these over a period and across different learners can generate data for educational analytics. Future work can conduct an analysis of these data to find patterns on how learners use and misuse axioms and theorems in coming up with their proofs. Similarly, data captured in the working memory that GPT uses when checking a learner's proof should also be stored and mined. Insights on common misconceptions can then be gleaned from these two data sources and subsequently be used by teachers in designing teaching materials to address difficulty that learners encounter in Geometry proving.

References

- Ainsworth, S. (2006). DeFT: A conceptual framework for considering learning with multiple representations. *Learning and instruction*, 16(3), 183-198.
- Baccaglioni-Frank, A., Antonini, S., Leung, A., & Mariotti, M. A. (2013). Reasoning by contradiction in dynamic geometry. *PNA*, 7(2), 63-73.
- Bass, L. E., Hall, B. R., Johnson, A., Wood, D. F., & Bess, S. W. (2001). *Prentice Hall Geometry: Tools for a Changing World*. Prentice Hall.
- Cinderella. (2015). Retrieved from <https://www.cinderella.de>
- Crowley, M. L. (1987). The van Hiele model of the development of geometric thought. *Learning and teaching geometry, K-12*, 1-16.
- The DOST Courseware Project. (2015). Retrieved from <http://courseware.dost.gov.ph/>
- Duval, R. (1998). Geometry from a cognitive point of view. *NEW ICMI STUDIES SERIES*, 5, 37-51.
- Fluckiger, J., Vigil, Y. T. Y., Pasco, R., & Danielson, K. (2010). Formative feedback: Involving students as partners in assessment to enhance learning. *College teaching*, 58(4), 136-140.
- Gantert, A. X. (2008). *Amsco's Geometry*. NYC: Amsco School Publications, Inc.
- GeoGebra. (2015). Retrieved from <https://www.geogebra.org/>
- Geometer's SketchPad. (2014). Retrieved from <https://www.dynamicgeometry.com>.
- GeoProof. (2015). Retrieved from <http://home.gna.org/geoproof/>
- Matsuda, N., & VanLehn, K. (2005, July). Advanced Geometry Tutor: An intelligent tutor that teaches proof-writing with construction. In *AIED*, vol. 125, pp. 443-450.
- Nyquist, J. B. (2003). The benefits of reconstruing feedback as a larger system of formative assessment: A meta-analysis. Unpublished Master's Thesis, Vanderbilt University.
- Regalado, R. V., Boñon, M. L., Chua, N., Piñera, R. R., & Cruz, S. R. D. (2015). Salinlahi III: An Intelligent Tutoring System for Filipino Language Learning. *ACL-IJCNLP 2015*, 87.
- Wong, W. K., Yin, S. K., Yang, H. H., & Cheng, Y. H. (2011). Using Computer-Assisted Multiple Representations in Learning Geometry Proofs. *Educational Technology & Society*, 14(3), 43-54.