Matsuda, N., & VanLehn, K. (2005, to appear). Advanced Geometry Tutor: An intelligent tutor that teaches proof-writing with construction. In *Proceedings of the 12th International Conference on Artificial Intelligence in Education*.

Advanced Geometry Tutor: An intelligent tutor that teaches proof-writing with construction

Noboru Matsuda[*1] and Kurt VanLehn[*2]
mazda@cs.cmu.edu vanlehn@cs.pitt.edu

[*1]*Human-Computer Interaction Institute, Carnegie Mellon University*
[*2]*Learning Research and Development Center, University of Pittsburgh*

Abstract: Two problem solving strategies, forward chaining and backward chaining, were compared to see how they affect students' learning of geometry theorem proving with construction. In order to determine which strategy accelerates learning the most, an intelligent tutoring system, the Advanced Geometry Tutor, was developed that can teach either strategy while controlling all other instructional variable. 52 students were randomly assigned to one of the two strategies. Although computational modeling suggests an advantage for backwards chaining, especially on construction problems, the result shows that (1) the students who learned forward chaining showed better performance on proof-writing, especially on the proofs with construction, than those who learned backward chaining, (2) both forward and backward chaining conditions wrote wrong proofs equally frequently, and (3) the major reason for the difficulty in applying backward chaining appears to lie in the assertion of premises as unjustified propositions (i.e., subgoaling).

1 Introduction

Geometry theorem proving is one of the most challenging skills for students to learn in a middle school mathematics [1]. When a proof requires construction, the difficulty of the task increases drastically, perhaps because deciding which construction to make is an ill-structured problem. By "construction," we mean adding segments and points to a problem figure as a part of a proof. Our hypothesis is that teaching a general strategy for solving construction problems should help student acquire the skill, and that teaching a more computationally effective problem solving strategy might elicit faster learning.

For theorem proving that does not require construction, there are two common problem solving strategies: forward chaining and backward chaining. *Forward chaining* (FC for short) starts from given propositions and continuously applies postulates [1] forwards, that is, by matching the postulates' premises (antecedents) to proved propositions and instantiating its conclusions as newly proved propositions. This continues until FC generates a proposition that matches the goal to be proved. *Backward chaining* (BC for short) starts from a goal to be proved and applies postulates backwards, that is, by matching a conclusion of the postulate to the goal, then posting the premises that are not yet proved as new goals to be proved.

In earlier work [2], we found a semi-complete algorithm for construction that is a natural extension of backwards chaining, a common approach to proving theorems that do not involve construction. The basic idea is that a construction is done only if it is necessary for applying a postulate via backwards chaining. The same basic idea can be applied to the FC strategy.

We have conjectured that both BC and FC versions of the construction strategy are comprehensible enough for students to learn. A question then arises: would FC or BC better facilitate learning geometry theorem proving with construction? Furthermore, if there is any difference in the impact of different proof strategies, what would it be? This study addresses these questions.

[1] In this paper, a geometric "postulate" either means a definition, an axiom, or a proven theorem.

Earlier work suggests that there are pros and cons to both FC and BC as vehicles for learning proof-writing. From a cognitive-theories point of view, some claim that novice students would find it difficult to work with backward chaining [3, 4]. But others claim that novice to expert shift occurs from BC to FC [5, 6]. From a computational point of view, we found that FC is more efficient for theorem proving *without* construction, but BC is the better strategy for theorem proving *with* construction [2]. Yet we are lacking theoretical support to determine which one of these strategies better facilitates learning proof-writing with construction.

To answer the above questions, we have built two versions of an intelligent tutoring system for geometry theorem proving with construction, called the Advanced Geometry Tutor (AGT for short). The FC version teaches the construction technique embedded in forward chaining search. The BC tutor teaches the construction technique embedded in backward chaining search. We then assigned students to each tutoring condition, let them learn proof-writing under the assistance of AGT, and compared their performance on pre- and post-tests.

In the remaining sections, we first provide a detailed explanation of AGT. We then show the results from the evaluation study. We then discuss lessons learned with some implications for a future tutor design.

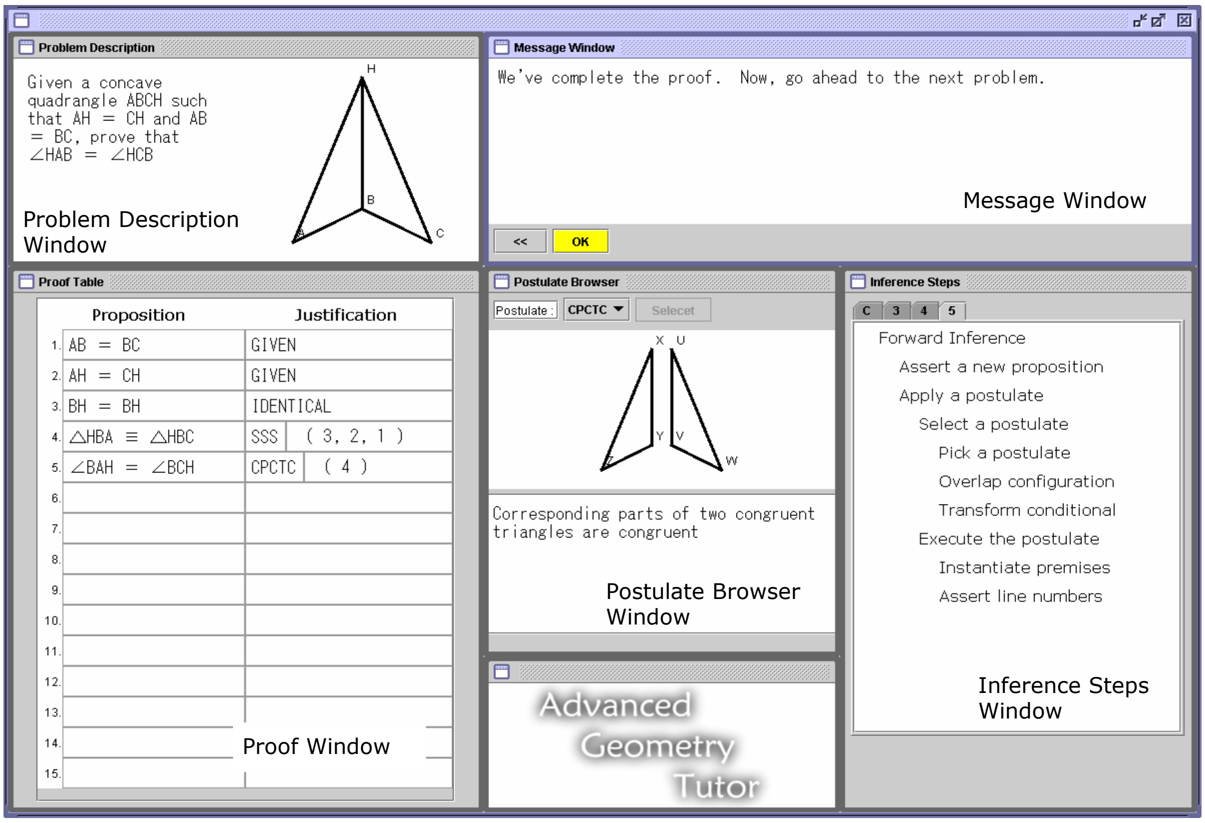
2 Advanced Geometry Tutor

This section describes the architecture of AGT. We first introduce the AGT learning environment. We then explain the scaffolding strategy implemented in AGT.

2.1 AGT learning environment

As shown in Figure 1, AGT has five windows each designed to provide a particular aid for learning proof writing.

Problem Description window: This window shows a problem statement and a problem figure. The problem figure is also used for construction. That is, the student can draw lines on the problem figure when it is time to do so.



Given a concave quadrangle ABCH such that AH = CH and AB = BC, prove that ∠HAB = ∠HCB

We've complete the proof. Now, go ahead to the next problem.

	Proposition	Justification	
1.	AB = BC	GIVEN	
2.	AH = CH	GIVEN	
3.	BH = BH	IDENTICAL	
4.	△HBA ≡ △HBC	SSS	(3, 2, 1)
5.	∠BAH = ∠BCH	CPCTC	(4)
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			

Corresponding parts of two congruent triangles are congruent

- Forward Inference
  - Assert a new proposition
  - Apply a postulate
    - Select a postulate
      - Pick a postulate
      - Overlap configuration
      - Transform conditional
    - Execute the postulate
      - Instantiate premises
      - Assert line numbers

Figure 1: Advanced Geometry Tutor

Proof window: Although there are several ways to write a proof, we focus on a proof realized as a two-column table, a standard format taught in American schools, where each row consists of a proposition and its justification. A justification consists of the name of a postulate and, if the postulate has premises, a list of line numbers for the propositions that match its premises. The Proof window shown in Figure 1 shows a complete proof for the problem in the Problem Description window.

Message window: All messages from the tutor appear in this window. When the tutor provides modeling (explained in Section 2.2), the instructions that a student must follow appear here. When a student makes an error, feedback from the tutor also appears here. More importantly, this window is used for the students' turn in a tutoring dialogue, which sometimes consists of merely clicking the [OK] button. The dialogue history is stored, and the student is free to browse back and forth by clicking a backward [<<] and a forward [>>] button.

Postulate Browser window: The student can browse the postulates that are available for use in a proof. When the student selects a postulate listed in the browser's pull down menu, the configuration of the postulate, its premises, and its consequence are displayed. This window is also used by the tutor. As shown in Figure 1, when the tutor provides scaffolding on how to apply a particular postulate to a particular proposition, the configuration of the postulate changes its shape so that the student can see how the postulate's configuration should be overlapped with the problem figure.

Inference Step window: Although applying a postulate may seem like a single step to an expert, for a novice, it requires following a short procedure. The Inference Step window displays this procedure as a goal hierarchy of indented texts where each line corresponds to a single inference step in the postulate application procedure. The tutor highlights the inference step that is about to perform. The Inference Step window in Figure 1 shows inference steps performed to fill in the 5th row in the proof table.

2.2 *Scaffolding strategy*

The tutor uses both proactive and reactive scaffolding. Proactive scaffolding occurs before the step it addresses, whereas reactive scaffolding (feedback) occurs after the step.

To adapt the level of proactive scaffolding to the student, we apply Wood, Wood and Middleton's tutoring strategy [7], where the rule is, "If the child succeeds, when next intervening, offer less help; If the child fails, when next intervening, take over more control." The student's competence level for a step is maintained as follows. When the student correctly performs a step, the tutor increases the competence level. Conversely, when the student commits an error on a step, then the competence level for that step is decreased. Based on the student's competence level for a step, the tutor selects one of three types of proactive scaffolding: **Show-tell**: the tutor tells students what to do and actually performs the step. **Tell**: the tutor tells students what to do, but asks the student to perform the step. **Prompt**: the tutor only prompts the student to perform the step.

Reactive scaffolding (feedback) occurs immediately after a step. On the first failure to enter the step, the tutor provides minimal feedback (e.g., "Try again"). If the student fails again to enter this step, the tutor's help varies according to the student's competence level. For example, for an inference step for construction the tutor would say "Draw segments so that the postulate has a perfect match with the problem figure." When the student still fails to draw correct segments, the tutor lowers the competence level of that inference step and then provides a "Tell" dialogue, which generates a feedback message like "Draw new segments by connecting two points." If the students still can not make a correct construction, then the tutor provides more specific "Show-Tell" dialogue that would say "Connect points A and B." Note that this sequence roughly corresponds to a sequence of hints that starting from a general idea and becoming more concrete until very specific instruction (a bottom-out hint).

The tutor only gives hints when the student has made mistakes. Unlike many other tutors, AGT has no “Hint” button that students can press when they are stuck and would like a hint. However, the tutor does act like other tutors in keeping the student on a solution path. For instance, when there are several applicable postulates, the tutor will only let the student choose one that is part of a correct proof.

Although we chosen these instructional policies based on pilot testing and personal experience in tutoring geometry students, and we believe that they are appropriate for this task domain and these students, we have not compared them to other policies. Indeed, they were held constant during this study so that we could fairly evaluate the learning differences caused by varying the problem solving strategy that the tutor taught.

3 Evaluation

An evaluation study was conducted in the spring of 2004 to test the effectiveness of AGT and to examine an impact of different proof strategies on learning proof writing.

3.1 Subjects

52 students (24 male and 28 female) were recruited for monetary compensation from the University of Pittsburgh. The average age of the students was 23.3 (SD = 5.4). The students were randomly assigned to one of the tutor conditions where they used AGT individually.

3.2 Procedure and materials

Students studied a 9-page Geometry booklet, took a pre-test for 40 minutes, used an assigned version of AGT to solve 11 problems, and took a post-test for 40 minutes. Detailed explanations follow.

The booklet described basic concepts and skills of geometry theorem proving. It contained (1) a review of geometry proofs that explains the structure of geometry proofs and the way they are written, (2) a technique for making a construction, and (3) explanations of all 11 postulates used in the study. For each postulate, the booklet provided a general description of the postulate in English, a configuration of the postulate, a list of premises, and the consequence of the postulate. The booklet was available throughout the rest of the experiment, including all testing and training.

Pre- and post-tests consisted of three fill-in-the-blank questions and three proof-writing questions. The fill-in-the-blank questions displayed a proof-table with some justifications left blank and asked students to supplement those blanks. The proof-writing questions provided students with a proof table that was initialized with either a goal to be proven (for the FC condition) or given propositions (for the BC condition). There was one problem that did not require construction and two that required construction.

For both tutoring conditions, two tests, Test-A and Test-B, were used for the pre- and post-test. Their use was counterbalanced so that the half of the students took Test-A as a pre-test and Test-B as a post-test whereas the other half were assigned in a reversed order. Test-A and Test-B were designed to be isomorphic in the superficial feature of the questions and their solution structures, as well as the order of the questions in the test. Our intention was that working the tests would require applying exactly the same geometry knowledge in exactly the same order.

Besides the six problems used in the pre- and post-tests, 11 problems were used during the tutoring sessions. Among the 11 training problems, six required construction that could be done by connecting existing two points.

3.3 Results

A post evaluation analysis revealed that question 5 (a proof-writing problem) in Test-A and Test-B were not exactly isomorphic; question 5 in Test-B required additional application of

CPCTC (the Corresponding Part of Congruent Triangles are Congruent postulate) and SSS (the Side-Side-Side triangle congruent postulate). The students who took Test-B made more errors than those who took Test-A on question 5 hence there was a main effect of the test version on the pre-test: $t(50) = 2.32$; $p = 0.03$. When we excluded question 5 from both Test-A and Test-B, the main effect disappeared. Hence the following analyses exclude question 5 from both pre- and post-tests unless otherwise stated.

To evaluate an overall performance on the pre- and post-test, we used following variables to calculate individual students' post-test score. For fill-in-the-blank questions, the ratio of the number of correct answers to the number of blanks was calculated. For proof-writing questions, the ratio of correct proof statements to the length of a correct proof was calculated.

With these scores, students using the FC version of the tutor performed reliably better on the post-tests than students using the BC version. In an ANOVA, there was a main effect for the tutor on the post-test: $F(1,48) = 10.13$; $p<0.01$. The regression equation of the post-test score upon the pre-test score and the tutor condition was: Post-test = 0.52 * pre-test – 0.14 (if BC) + 0.50. Using the pre-test scores as a covariate in an ANCOVA, the adjusted post-test scores of 0.58 and 0.72 for the BC and FC students were reliably different. The effect size[2] was 0.72. In short, the FC students learned more than the BC students by a moderately large amount.

To see how the FC students outperformed the BC students, we conducted an item analysis by comparing scores on the fill-in-the-blank and proof-writing questions separately. For fill-in-the-blank questions, there were no significant differences between FC and BC students on the pre-test scores nor on post test scores. However, there was a main effect of the test (i.e., pre vs. post) on test scores for both FC and BC students: *paired-t*(25) = 2.74; $p = 0.01$ for FC, *paired-t*(25) = 3.43; $p < 0.01$ for BC. That is, both FC and BC students performed equally well on fill-in-blank questions, and they improved their performance equally well.

On proof-writing questions, the difference in pre-test was not significant ($t(50) = 0.91$; $p = 0.37$), but there was a main effect of tutor conditions for the post-test scores: $t(50) = 2.53$; $p = 0.02$. The effect size was 0.93.

The difference in the overall post-test scores between BC and FC students was thus mainly from the difference in proof-writing questions: the FC students wrote better proofs than BC students on the post-test. To understand how the FC students outperformed the BC students in proof writing, we further compared their performance on proof-writing with and without construction.

Since we excluded question 5, which was a construction problem, there was only one non-construction problem (question 4) and one construction problem (question 6). Figure 2 shows mean scores on these questions. The difference in the non-construction problem was not significant: $t(50) = 0.66$; $p = 0.51$, whereas the difference in the construction problem was significant: $t(50) = 2.89$; $p < 0.01$. That is, FC and BC students tied on non-construction problem, but FC students outperformed BC students on construction problem.

In order to narrow the locus of difference even further, we conducted 3 further analyses of the superiority of FC to BC. The analyses contrasted (1) the type of proof

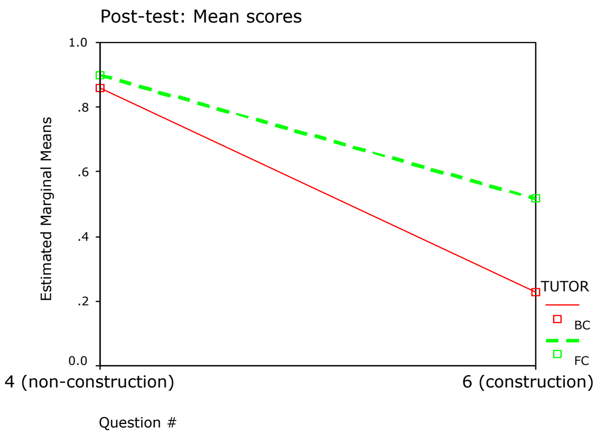


Figure 2: Mean scores on proof-writing for problem with and without construction

[2] A ratio of the difference between FC and BC mean adjusted post-test scores to the standard deviation of the BC pre-test scores.

written for each problem, (2) the types of proof statements appeared in each proof, and (3) the quality of postulate applications used to compose each proof statement.

Before discussing these analyses, we need to introduce the scheme used to code proof statements. A proof statement, which is written on a single row in the proof table, consists of a proposition, a justification, and premises. A proof statement is said to be *on-path* when it is a part of a correct proof. An *off-path* proof statement is not a part of a correct proof, hence its proposition may or may not true, but the postulate used as a justification has a consequence that unifies with the proposition, and its antecedents unify with the premises listed in the justification. A *wrong* proof statement is neither on-path nor off-path.

Figure 4 shows the number of occurrence of each type of proofs. "OD" shows the number of proofs that were not written in the strategy taught (called TStrategy). The rest of this section excludes OD proofs. The figure clearly shows that FC students wrote more *correct* proofs, which by definition contain a tree of on-path proof statements connecting the givens to the top goal. FC and BC students were equally likely to write *wrong* proofs, which contains a tree of proof statements but the tree involves at least one proof statement that is not on-path. Aggregating *stuck* proofs where a proof does not contain a tree of proof statement, and *blank* proofs where no attempt for proof was made at all, BC students were more likely than FC students to fail in these ways.

Moving now to the statement-level analysis, there were 479 proof statements (215 and 264 in BC and FC conditions) appearing on the post-test. Of those, 400 were *reasonable* (i.e., either on-path or off-path) and 79 were *wrong* statements. 180 statements (92 in BC and 88 in FC) were *missing*, which means that they are necessary for a correct proof but were not mentioned at all. Figure 3 shows the frequency of each type of proof statements.

GRAMY often made off-path statements, especially when using FC to do constructions. However, the students seldom made off-path statements, especially in correct proofs, where only 3 off-path statements were written by FC students and no off-path statements were written by BC students. In incorrect proofs, off-path statements were slightly more frequent (19 for FC; 7 for BC), and FC students wrote more off-path proof statements than BC student ($\chi^2 = 8.52$; $df = 1$; $p < 0.01$). A further analysis revealed that all those off-path statements were made for postulate applications that did not involve construction. That is, when they made a construction, the students always write an on-path proof statement.

Another interesting phenomenon that can be read from Figure 3 is that BC students wrote wrong proof statements more frequently than FC students. Together with the fact mentioned earlier that the BC students tended to fail to start writing a proof (i.e., the Blank proofs in Figure 4), BC students apparently found it more difficult to write reasonable proof statements (on- and off-path) than FC students.

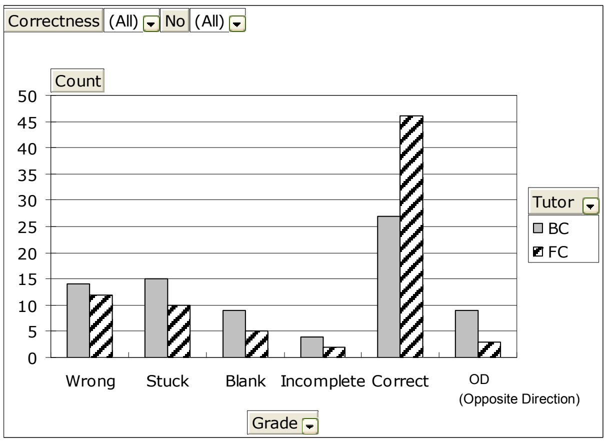


Figure 4: Classification of proofs

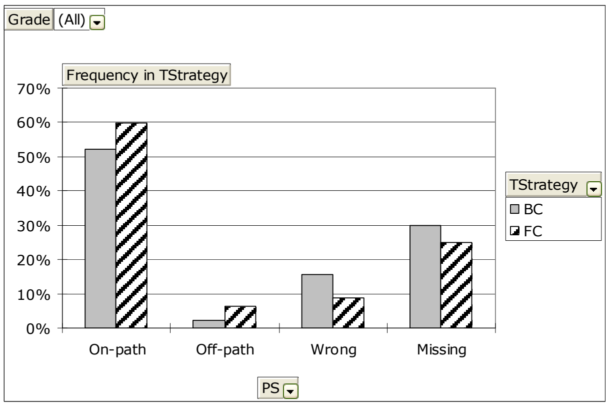


Figure 3: Classification of proof statements

As for the analysis on the quality of postulate applications, to investigate a reason for BC students having difficulty on writing proof statements, we coded each of the 79 wrong proof statements as a triple of independent codes of (1) the proposition, (2) the justification, and (3) the premises, which are three constituents of a proof statement. For each proof statement, we coded each instance of these constituents as on-path, off-path, wrong, or blank. We then ran 2 x 4 Contingency table analyses on each constituent to see if there was difference in the frequency of these constituents between BC and FC students.

For propositions and justifications, FC and BC did not display different frequency distributions. There was, however, a significant difference in the use of premises between FC and BC students. Figure 5 shows a 2 x 4 Contingency table on the use of premises. A Fisher's exact test on the table was 7.25 ($p = 0.04$), indicating a significant difference in the distribution of codes for premises. The BC students tended to leave the premises blank more often than the FC students. This tendency of leaving the premise blank was one reason for the inferiority of BC students in writing correct proofs compared to the FC students.

			Premises				Total
			Blank	Off-path	On-path	Wrong	
TStrategy	BC	Count	27	2	1	18	48
		Expected Count	21.9	3.6	.6	21.9	48.0
	FC	Count	9	4	0	18	31
		Expected Count	14.1	2.4	.4	14.1	31.0
Total		Count	36	6	1	36	79
		Expected Count	36.0	6.0	1.0	36.0	79.0

Figure 5: A 2 x 4 Contingency table on the use of premises

4 Discussion and Concluding Remarks

4.1 Learning proof-writing with construction

The first major contribution of this study is showing that proof-writing with construction can be taught with a technique that is a natural extension of theorem proving without construction. Although geometry construction is a difficult skill, perhaps even a creative one, it can be taught by conventional ITS technology, given that the tutor has an explicit problem solving strategy to teach that will solve construction problems.

Although there was not a main effect on accuracy of postulate applications measured with the fill-in-the-blank questions on the post test, some students (the FC ones) outperformed other (the BC ones) in proof-writing. This suggests that understanding domain principles (i.e., the concept of geometric postulates) is not sufficient for writing correct proofs. In addition, one must acquire proof-writing skills, and different kinds of instruction are differentially effective at facilitating this.

4.2 Impact of the different proof strategies on learning proof-writing

Despite the much higher computational demands of the FC version of the construction algorithm compared to the BC version, as documented in computational experiments with GRAMY [2], it turned out that FC students acquired more skill at construction than BC students. Our finding agreed with other empirical studies showing novice students' difficulty in applying backward chaining. It seems that problem solving complexity for a computer does not necessarily imply learning complexity for humans. Indeed, although both GRAMY and the students used both FC and BC, GRAMY always produces many off-path proof statements whereas the humans rarely did. This suggests that the humans are using knowledge or strategies not represented in GRAMY.

4.3 Difficulty in subgoaling

The BC students tended to get stuck at providing premises even when they picked a correct proposition and a postulate. It seems to be difficult for BC students to specify *subgoals* as the to-be-justified propositions that support a postulate application.

Subgoaling requires that the students write into the table one or more propositions (i.e., to satisfy the premises of a justification) that have yet to be proved. At the time they are entered into the proof table, those premises are not "true" assertions, but just hypothesis to be proved. This uncertainty may increase the chance of failure in backward chaining. Furthermore, those propositions are usually new in the proof table. Forward chaining, on the other hand, always enters propositions that are derived from known facts. Backward chaining differs mostly from forward chaining in this guess-and-try fashion in entering proof statements.

4.4 Implications for a future tutor design

A potential way to improve the BC tutor's efficacy is to intensify modeling and scaffolding on subgoaling for backward chaining. Although asserting unjustified propositions into a proof step was explicitly stated in the cognitive model of backward chaining utilized in AGT, the model was not effective in supporting the BC students in learning subgoaling.

The inadequacy of the BC tutor may also be due to a lack of instruction on *backtracking*. Backward chaining is essentially nondeterministic. For some goals, there are multiple equally plausible postulates whose consequences unify with the goal. Therefore, one must choose one of the postulates, try it, and if it does not work well, back-up to the choice point and choose another postulate. AGT acted as a more restricted tutor. Instead of allowing students to choose a postulate and possibly backup to this choice later, the tutor only allows them to choose an on-path postulate, so they never had to back up during training. This design principle is supported by an observation that the more the students flounder, the less opportunity they have for each cognitive skill to be exposed hence they achieve less learning [8]. For subgoaling, however, it might be necessary for students to understand that they are asserting hypotheses that could be wrong. Moreover, when applying backward chaining during the post-test, students may have had to choose among equally plausible postulates. This could cause confusion and consternation. Thus, it might be necessary to let students backtrack during training.

A related issue is to teach students to recover when they get stuck. Since the backward chaining strategy may lead them to an impasse, they should be taught what to do when they get stuck. AGT did not do this. Perhaps that is why the BC students often got stuck during the post-tests. AGT should train an ability to analyze the situation to identify an impasse, to diagnose the cause of the impasse, and to figure out an alternative way to avoid it by selecting a different path.

Acknowledgements: This research was supported by NSF Grants 9720359 and SBE-0354420.

Reference:

1. Senk, S.L., *How well do students write geometry proofs?* Mathematics Teacher, 1985. **78**(6): p. 448-456.
2. Matsuda, N. and K. VanLehn, *GRAMY: A Geometry Theorem Prover Capable of Construction.* Journal of Automated Reasoning, 2004. **32**(1): p. 3-33.
3. Trafton, J.G. and B.J. Reiser, *Providing natural representations to facilitate novices' understanding in a new domain: Forward and backward reasoning in programming.* Proceedings of the 13th Annual Conference of the Cognitive Science Society, 1991: p. 923-927.
4. Anderson, J.R., F.S. Bellezza, and C.F. Boyle, *The Geometry Tutor and Skill Acquisition*, in *Rules of the mind*, J.R. Anderson, Editor. 1993, Erlbaum: Hillsdale, NJ. p. 165-181.
5. Larkin, J., et al., *Expert and Novice Performance in Solving Physics Problems.* Science, 1980. **208**(4450): p. 1335-1342.
6. Chi, M.T.H., P.J. Feltovich, and R. Glaser, *Categorization and representation of physics problems by experts and novices.* Cognitive Science, 1981. **5**: p. 121-152.
7. Wood, D., H. Wood, and D. Middleton, *An experimental evaluation of four face-to-face teaching strategies.* International Journal of Behavioral Development, 1978. **1**(2): p. 131-147.
8. Anderson, J.R., et al., *Cognitive tutors: Lessons learned.* Journal of the Learning Sciences, 1995. **4**(2): p. 167-207.