Modeling Hinting Strategies for Geometry Theorem Proving

Noboru Matsuda and Kurt VanLehn
Intelligent Systems Program
University of Pittsburgh
mazda@pitt.edu, vanlehn@pitt.edu

Abstract: This study investigates effective hinting strategies to help students learn geometry theorem proving. The fundamental motivation is to help students solve problems by themselves so that they can improve proof skills to the mastery level (i.e., can apply knowledge when appropriate). However, until they achieve mastery, the students frequently encounter impasses during problem solving. The tutor must provide help to overcome those impasses. The present study analyzes the kind of help that should be provided. A study was conducted with the students at intermediate skill level proving geometry theorems individually with a human tutor. Nearly all (97%) help events were targeted at a single step in a proof, i.e., a hint on a postulate application. We developed a language to describe such hints. The formal language for hints and the schematic hinting strategy we have observed facilitate building an intelligent tutoring system that provides "elicitative" help.

1. Introduction

There are many types of difficulties that students suffer when they learn geometry theorem proving (GTP). One common difficulty is that the students cannot build up a proof even when they "know" the skills for GTP including basic postulates and inference rules, i.e., forward and backward chaining. In other words, they might have acquired sufficient skills to produce a proof by the time they solve geometry problems, but they can not make use of those skills in an actual problem-solving situation. It is this class of students (the *intermediate* students) that we target to teach. The important issue for the intermediate students is to let them solve geometry problems by themselves to improve their proof skills to the mastery level.

It is known that both learning from worked-out examples and learning by solving problems are effective for acquiring problem-solving skills (Sweller & Cooper, 1985). The cognitive load theory advocates that the novice students must start with studying with worked-out examples, and shift to problem solving as they improve competence in the problem-solving skills (Kalyuga, Chandler, Tuovinen, & Sweller, 2001). Until the students achieve mastery, their problem-solving technique is inefficient, hence they make many inappropriate explorations, and more importantly, they frequently encounter impasses during problem solving. To overcome the impasse, the intermediate students ask the tutor "what to do next?" and receive a hint that reminds them of what they could have applied by themselves. The hint must be modest; if the tutor provides too much help the student might not strengthen the accessibility of their knowledge. On the other hand, if the tutor provides too little help, the student can not overcome the impasse. The amount of scaffolding provided by the tutor must be adapted to individual student's need.

Our goal is to build an intelligent tutoring system for GTP to help intermediate students learn proof skills by solving problems. Namely, we are seeking a technique to provide students with effective hints so that they can work through the problems by themselves. Two issues arise: how should the tutor provide hints for students to overcome their impasse, and how should the tutor deal with the students' unskilled inferences including wrong and unnecessary assertions?

Hinting is essential in intelligent tutoring systems (ITS). In fact, virtually all ITS have hinting as a part of their instructional tactics. There is little known, however, about effective hinting. The hints have been rather ad-hoc and stereotypic, e.g., providing an abstract hint first and gradually shifting to a specific one, ended up with providing a concrete answer (the so called bottom-out hint). Hume et al conducted an extensive analysis on hinting in human-to-human tutoring (Hume, Michael, Rovick, & Evens, 1996). Their theory of hinting categorized various hints in terms of the type, the structure, and the timing of hint. The theory of hinting could be a model of hinting schema for ITS. However, as Zhou et al (1999) claimed, the human tutors' hinting tactics are so subtle that the theory of hinting proposed by Hume et al does not fully explains how to produce a hint as flexible and context sensitive as the human tutors do. Zhou et al then implemented a hinting schema in their ITS (Circsim-Tutor) by categorizing students' incomplete answer and specifying a hinting dialogue for each incomplete answer. Yet, their hinting schema does not tell how to select an appropriate context of hinting.

Another fundamental design issue is to deal with students' unskilled inferences that are correct but do not follow either forward nor backward inference, or that do not even happen to participate in a solution. In one extreme, model tracing tutors sometimes demand that the students follow an ideal solution path, i.e., the model inference (Anderson, Boyle, & Yost, 1985). Other model tracing tutor accept any correct inference even if it is not on an ideal solution path (Koedinger & Anderson, 1993), but when a student reaches an impasse, the model tracing tutor provides a hint on the next step that is a strict backward or forward inference from the top-level goal or given, no matter what assertions the student has made so far. That is, even if the model tracing tutor simply "accepts" students' inappropriate assertions, those assertions would never be considered as a basis of hint. It is indeed often reported that intermediate students have difficulties to express their inference in the logical language (i.e., a form of proof) even when they "understand" a proof (see, for example, Dreyfus, 1999; Hoffer, 1981). Hence, it is controversial whether providing a less restricted and more student adapted learning environment might bring better learning gain than the model tracing tutors, for such an environment agrees more with the students' inference style (Corbett & Anderson, 1992).

In sum, we suspect that it is beneficial for students learning GTP to follow their habit of inference. That is, we hypothesize that learning by problem solving in less restricted learning environment with elicitative hinting will encourage the intermediate students to reach mastery-skill level faster than in a more restrictive environment. However, to explore this issue, we must first determine how to give effective hints in such an environment. For this, we turned to data from human tutoring.

We collected protocol data from middle school students working on GTP, and analyzed them to find a pattern of students' inferences as well as to find the tutor's hinting strategies. As we will discuss later (and is the major motivation of the current study), the analysis has revealed that the students follow neither a strict top-down nor a strict bottom-up strategy. That is, this natural strategy is not consistent with the problem-solving strategy taught in the model tracing tutors. This disagreement motivated us to design learning environment with more freedom on students' activities. As will be seen, offering more freedom on students' inference requires more intelligence in the tutor to monitoring students' proofs and providing appropriate hints. What we have learned from the data is that the human tutor's hinting strategy can be described as a hierarchical plan. The data have also revealed that nearly all (87 of 90; 97%) of the human tutor's hints can be described in a formal language with respect to focus and form of hint. The major contribution of the present paper is a proposed language to describe the hints and the automated hinting schema for a learning environment that allows students' opportunistic (or "unskilled," if you will) inferences.

The remaining sections first explain the study we conducted as well as the protocol data analyzed. We then discuss a typical opportunistic inference style of the middle school students that becomes a basis of a discussion on a difficulty to design an ITS. Next, we propose a language to describe an "elicitative" hint. Finally, we discuss a hinting strategy and implication for the automation of hinting schema for an intelligent tutoring system in GTP.

2. GTP Study

This section describes an experimental study by a human tutor and the middle school students learning GTP by solving geometry theorems.

2.1. Method

There were 8 students randomly selected in a Japanese middle school (we call them with the identification number 201 through 209 here after). Three geometry proof-problems P1, P2, and P4 were used in the study where P1 and P2 were assigned to all students, whereas P4 was assigned only when the situation permitted (i.e., the student's motivation, the amount of time spent, etc.). Problem P2 and P4 were construction problems, which require students to draw additional lines by compasses and straightedges to complete a proof.

Each student solved problems individually while thinking out aloud. The sessions were video taped and transcribed. Video taping is essential for this study, because the students often pointing the problem figure while speaking, say, "*these* two segments are equal."

In the study, the tutor was asked to provide hints only when the students couldn't otherwise precede a proof. Namely, the tutor provided a hint upon a student's request or when the tutor detected students' floundering. The human tutor was asked not to provide didactic help. Instead, he was supposed to elicit a piece of knowledge from students necessary to overcome the impasse.

2.2. Result

The student #201 solved all three problems without tutor's hint hence was omitted from the analysis. The student #204 and #206 could not solve any problem at all even with the tutor's hint hence was omitted from the analysis as well. The remaining 6 students solved 15 problems as shown in Table 1. The 6 students completed 10 problems with the human tutor's help. Tutor did not provide any hint in #209 on P1, #203 on P2, and #203 on P4, hence those sessions were not counted. Of the remaining 12 sessions, only in 2 sessions (#205 on P1, and #207 on P4) the tutor's hints did not help students to complete a proof; showing that the tutor's hints were indeed effective. The numbers in the successful 10 sessions show the number of hint events observed.

	P1		P2		P4	
202	Guided	1	Guided	3		
203	Guided	4	*Time up. Only 1 hint for construction prvided.*		*Time up. Only 1 hint for construction prvided.*	
205	*Guided from the beginning, but none of the hints worked.*		Guided	4		
207	Guided.	2	Assisted	4	*S couldn't understand a correct proof.*	
208	Guided	4	Guided	3		
209	*Time up. No hint provided*		Assisted	3	Guided	4

Table 1: Students' performance on each problem

The students' utterances in the transcriptions were segmented so that a single segment corresponds to a proof step or a response to the tutor's assistance. The tutor's utterances were segmented so that a single segment corresponds to a hint. The following sections show analysis of the protocol data.

3. Students' Inference in GTP

We first questioned how the students' proofs grow. To answer this question, we located individual student's utterances in a proof tree, and observed a pattern of progress in their proof. Our main concern is to see whether the students make forward and/or backward chaining inferences strategically.

3.1. Pattern of Students' Assertions

As an example, Figure 1 shows a chronological progress of student #205's reasoning, mapped onto a proof tree, observed until she reached an impasse. The goal to be proven is shown at the top of the tree, whereas the given at the bottom. A branching link shows a conjunctive justification. Nodes with a rectangle show the propositions that this student asserted. The numbers on their shoulder show the order of assertion. Since the proposition C is a premise for both A and G, the first assertion is located on two places.

As shown in the figure, this student built up a proof neither in a strict forward chaining nor in a strict backward chaining manner. Rather she seems to assert facts (i.e., propositions) that were eventually recognized. This opportunistic assertion is not peculiar to this particular student. All students participated in our study showed the same behavior.

3.2. Discussion: What Should the Learning Environment for GTP Look Like?

The above data show that the students who are strengthening their proof skills apparently do not always follow the forward and/or backward inference strategy. Basic question arises: to gain better learning, should we allow students to employ their own inference style, or should we restrict their activities only to forward or backward chaining?

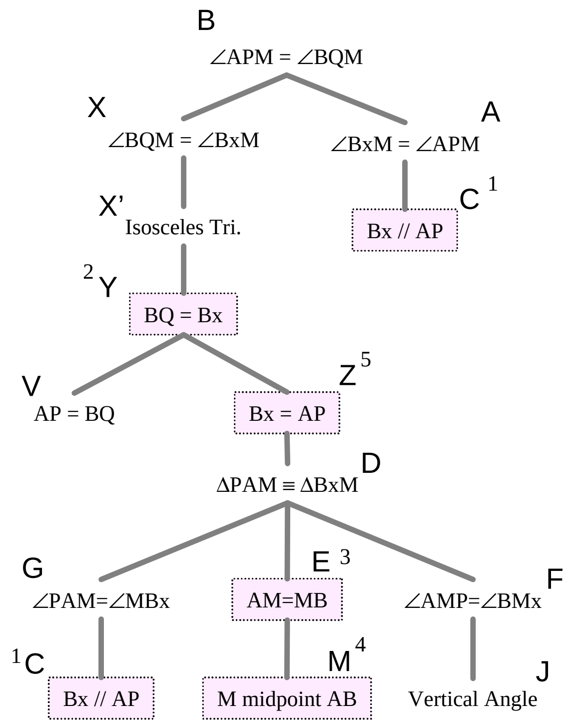

Figure 1: A typical progress of student's input over a proof tree

In one extreme, the model trace tutor tries to put a student back onto a right track when he/she deviates from a correct path or gets stuck. For instance, if the student #205 solving the problem shown in Figure 1 were facing an extreme model tracing tutor, the tutor would provide negative feedback on the student's second assertion (BQ=Bx), which has no direct connection to the first assertion (Bx//AP). This intervention could spoiled the further exploration made by the student. In even a less extreme model tracing tutor, if the student asked a "what's next" hint right after making the second assertion, the model tracing tutor would advise the student to consider the step C→G, which is the bottom most step with all the premises asserted. This hint completely ignores the student's first assertion.

On the other hand, the human tutor is quite patient with such scattered assertions. When a student gets stuck, the tutor provides a hint based on the student's effort up to that point. Such a hint might make more sense for students than the one based on a model reasoning, because it fits the students' current awareness of the problem state. It is not known, however, whether this student-centered learning environment is superior to the model tracing tutor. Hence, we are motivated to develop an ITS that allows students to build up a proof without any strict order, while taking those scattered assertions into account for hinting.

The next question, then, is how does the hinting in such an "unstructured" learning environment look like? Unlike more structured environment where the students share a problem-solving plan with the tutor, little is known about hinting in an unplanned situation (see VanLehn et al., 2002 for hinting with a shared problem-solving plan). To answer to this question, we further analyzed the protocol data to find out the human tutor's scaffolding pattern on hinting.

4. Ontology of Hint

We first analyzed the contents of hints that the human tutor provided. The tutor's utterances were segmented so that each segment corresponds to a single *hint*. In this study, a hint is defined as a tutor's contribution to a student's problem solving in a sense that it provides enough information for the students to remind them of what they could apply spontaneously. A sequence of hints was counted as a *hint event*. The tutor may have a plan for a hint event that leads a student to a particular state several steps away from the current state. A hint event may have a backup plan to recover a failure of the original plan.

4.1. The Observed Hint Events and Hints

We observed 31 hint events. They were categorized into 4 types of hint events; (1) a hint event for a next step, (2) a hint event for a justification, (3) a hint event for a construction, and (4) a hint event to get started a proof. Most of the hint events are aimed to help on a justification (14 of 31; 45%). This happened when a student mentioned a proposition but did not justify it. Ten of 12 (83%) of the hint events for a justification were forward chaining inference from the given premises to the proposition to be justified.

The second most frequently observed hint events are for a next step (10 of 31; 32%). This type of hint event was provided when a student could not make any progress on a proof. Since this type of hint looks more complicated than any other types of hint, the present paper is particularly focused on the next-step hint. As discussed below, there are several different ways to help the student find a way to get out of the impasse, depending on a status of the assertions made. The most fundamental and critical issue for this type of hint event is the selection of the step to hint. This is a tricky issue given that the tutor permits inferences to be done in any order (see section 5 for the details).

There were only 3 cases (10%) where the students could not find a construction and thus receive a hint. The hints were all straightforward, namely, the tutor simply provided a correct construction without any justification (or reason) of the construction. The human tutor reported afterwards that it seemed difficult to provide an indirect hint for construction. The fact that there are many unsuccessful constructions possibly exist supports the tutor's inability to find good explanation or hint for the construction. However, our analysis in the following section and a computational model of GTP with construction (Matsuda & VanLehn, 2000) suggests that there is a systematic way to provide hints for construction.

There was only 1 case (3%) where the student could not start with a proof at all and received a hint. Three hint events (10%) could not be classified.

4.2. Discussion: A Language of Hint

The next-step hint can either be about a general strategy (e.g., "Try to apply a postulate whose premises are all given"), or about an element of the proof itself, namely a proposition or a postulate application. We did not observe any general strategy hints. We organized the hints about propositions and postulate applications into a Cartesian product with respect to focus and form of hint.

The individual *hints* in those 25 hint events falls into one of four categories regarding the *focus* of hint: (1) a hint on a whole application of a postulate (e.g., “Remember if two sides of triangle are equal, then the base angles are also equal”), (2) a hint on a conclusion of a postulate application (e.g., “What can you conclude about the base angles in a triangle with two equal sides?”), (3) a hint on a premise of a postulate application (e.g., “If you want to prove these two angles are equal, what should be true among these two segments?”), and (4) a hint on a proposition apparently involved in a postulate application but not mentioning it explicitly (e.g., “Can you say anything about these two segments?”).

We observed five different *forms* of hint; (1) direct exhibition, (2) question asking a whole proposition, (3) question asking a relationship in the proposition, (4) question asking an element involved in a proposition, and (5) highlighting (or, pointing out) a related configuration in the problem figure. As an illustration, Table 2 shows all possible hints for a proof step that invokes the theorem of isosceles triangle (i.e., if two sides of a triangle are equal, then the base angles are also equal).

X->Y {X/a, Y/b}	(E) Exhibit	Question			(P) Pointing
		(W) Q whole statement	(R) Q on relation	(O) Q on object	
(AW) Whole application	If AB=AC, then <ABC=<ACB	What can you do now?	Can you say anything about segments AB and AC, and angles <ABC and <ACB?	-	Look at this triangle (^ABC)
(AP) Premise of application	It is sufficient to show AB=AC to conclude <ABC=<ACB	What should you prove when you want to conclude <ABC = <ACB?	You want to conclude <ABC=<ACB. Now, what should be true among AB and AC?	Which two segments must be equal to conclude <ABC=<ACB?	-
(AC) Conclusion of application	Given that AB=AC, <ABC and <ACB are equal.	What can you conclude when AB and AC are equal?	We know AB=AC. So, what can we conclude with <ABC and <ACB?	We know AB=AC. So, which two angles can you conclude to be equal?	-
(PP) Perceive Proposition	AB and AC are equal <ABC and <ACB are equal	What is known?	can you say anything about AB and AC (<ABC and <ACB)?	Which segment is equal to AB? Which angle is equal to <ABC?	Look at AB and AC (<ABC and <ACB)

Table 2: The type of hints for a next-step hint

In sum, when helping a student working on theorem proving, the tutor almost always provided a hint on a postulate application targeting a single step in the proof. A hint can vary in its focus and form.

We then classified the hints with the coding schema shown in Table 2. The codes express a focus and a form of hint. For example, a direct exposure (E) of a premise of postulate application (AP) is coded as AP-E. Table 3 shows the number of hints observed for each type of hint appeared in the hint events for next-step. Note that the 25 *hint events* mentioned here are composed of the 87 *hints* shown in Table 3.

The most frequently used hint is PP-R, which is asking a relationship between particular geometric elements involved in a target proposition. A typical utterance for this hint is “so, these two segments are…?” which is categorized as ‘Prompting’ in Person et al (2001). This type of hint was often used as the first trial of a hint with the focus on a proposition, as opposed to a postulate application. If this hint failed, then the tutor changed the focus of hint from the proposition to a postulate application with the original proposition as its conclusion (for a hint in backward chaining) or premise (for a hint in forward chaining), or simply provided PP-E hint on the same proposition (i.e., a bottom-out hint in a model tracing tutor).

	E	Question			P	Total
		W	R	O		
AW	3	0	0	0	16	19
AP	2	4	0	2	0	8
AC	1	6	5	0	0	12
PP	21	0	24	0	3	48
Total	27	10	29	2	19	87

Table 3: Frequency of hints observed in the next-step hint events

The second most frequently used hint is PP-E, which is a direct exposure of the target proposition. This hint was used not only to assert a proposition that a student could not mention, but also to draw the student's attention to a proposition that the student had already asserted as a preparatory step for a planed hint sequence (e.g., "Remember these two segments are equal. So, now …").

The third most frequently used hint is AW-P, which is highlighting a part of the problem figure that corresponds to a particular postulate application with an utterance, such as, "look at this triangle." The first hint given in the next-step hint event is most likely (7 of 10; 70%) AW-P for a step with all or all-but-one premises asserted.

In sum, we found that 87 out of 90 (97%) hints that the human tutor provided during the tutoring sessions fall into the hint category shown in Table 2. This table can be generated automatically for each step in the proof. Hence a hinting schema should consist of two steps: to select a target step, and to select an instance of a hint from the table.

5. Hinting Strategy for Next-step Hint

The tutor's strategy for selecting the target steps of a hint depended on the type of hint event. For justification hint events, the tutor simply hinted the postulate application whose conclusion was the to-be-justified proposition. On the other hand, selection of a target for a hint for a next-step hint event is more elaborated. Since the student could potentially do any inference next, how does the tutor choose which proof step to hint? Furthermore, how the tutor chooses a sequence of hints leading up to the target?

Due to the limited space of the paper, we only focus on the first issue, namely the target selection in the remaining section. The following observation gives us an insight into the second issue: once the target of hint is determined, the tutor makes a plan of hint events in a hierarchical decomposition fashion. That is, the ultimate goal of hinting is decomposed into several subgoals, each of which then becomes a target of hinting. If a plan failed, then a backup plan is considered. Further analysis of this planning strategy will be reported elsewhere.

5.1. Choosing the Target Step for a Next-step Hint Event

Based on the observed fact that the human tutor always provides a hint on a single proof step, we presuppose that the selection can be modeled as a process to identify an inference step in the proof tree with respect to a status of assertions located over the proof tree. By an inference step we mean a single subtree consisting of a premise(s), a conclusion, and a link between them representing a justification of the conclusion. Depending if these propositions and the justification are asserted (either by the student or the tutor) or not, there are 6 possible states of a subtree as shown in Figure 2.

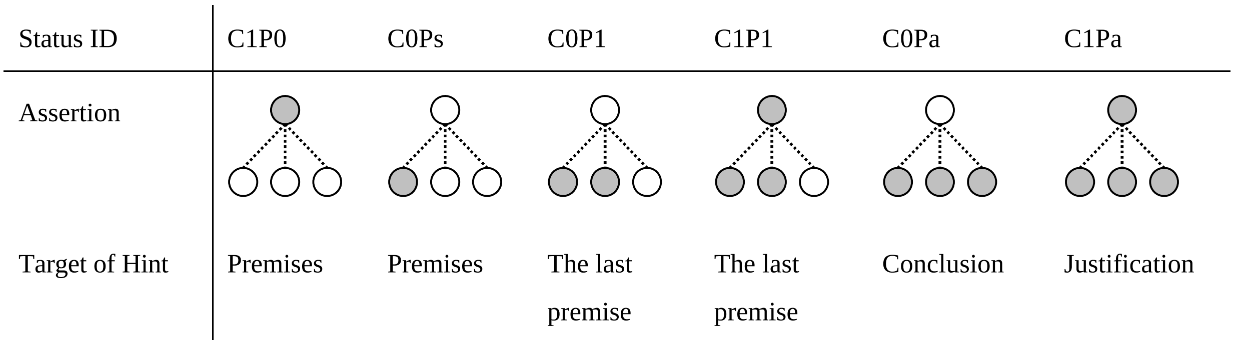

Figure 2: The subtree model for motivation of hinting

A status ID holds the information regarding the assertions made either by the student or the tutor. The first two letters of the ID show whether a conclusion is asserted (C1) or not asserted (C0). The following two letters show whether all the premise are asserted (Pa), none asserted (P0), only some of the premise are asserted (Ps), or all but one premises are asserted (P1). A gray and a white node in a subtree show a proposition that is asserted and unasserted respectively. The dotted links show a justification that is not mentioned.

To see how the human tutor determined a target step for a next-step hint event, we counted the number of subtree-types chosen as the target in 10 of the next-step hint events mentioned in 4.1. We also counted the number of all subtree-types that occurred at the time the next-step hint events were planned (i.e., at the time the students were stuck) in all the 10 next-step hint events. Table 4 shows those numbers.

	C0P1	C0Pa	C1Pa	C1P1	C1P0	C0Ps
Occurence	14	10	4	3	3	1
Choice	8	2	0	0	0	0
Ratio Choice/Occurence	57.1%	20.0%	0%	0%	0%	0%
# Fringe	2	1				

Table 4: Frequency of motivation of hinting in the 'next step' hint events

Interestingly, the subtrees C0P1 and C0Pa were common among the proof tree when the students got stuck. This means, there is almost always a subtree(s) that has all or all-but-one premises asserted by a forward inference. The reason for this phenomenon are not fully understood, but we consider the following fact as a support: 27 of 30 (90%) of the subtrees appeared in the problems used in the study have 1 or 2 premises, hence asserting a proposition most likely makes a subtree either C0P1 or C0Pa. The human tutor always chose either C0P1 or C0Pa as a target of next-step hint event. C0P1 is preferred; 57% of the C0P1 were targeted, whereas only 20% of C0Pa were selected as the target step.

5.2. Discussion: Towards an Automation of Hinting

The Table 4 also shows the number of the target subtrees that are the “fringe.” A fringe is a subtree that is either the most lately asserted proposition made by a complete forward chaining from the given, or the most lately assumed subgoal made by a continuous backward chaining from the top-level goal. In other words, a fringe in forward chaining has a complete justification, whereas a fringe in backward chaining is motivated by the top-level goal. Since the model tracing tutor utilizes a production system, which is a collection of production rules as a domain model, to determine the target step for next-step hint, it always provides a hint on the fringe (which is the only legal move to make next). Our study shows that it is uncommon for a human tutor to provide a hint on the fringe when the student makes an opportunistic inference. As shown in Table 4, only 3 of the 10 hints targeted postulate application on the fringe.

The above observations convince us that a hint schema for a next-step hint event can be automated. First, the target of the hint, namely, a proof step to work on next, can be determined with respect to a status of assertions in subtree. The human tutor’s preference on C0P1 and C0Pa as a target step might work well, because those are a part of proof that the student would have been paying attention most. What if there are more than one C0P1’s and C0Pa’s in the proof tree? – Further investigation is required for resolution of this issue.

6. Conclusion

The analysis of protocol data gathered from the student in middle school has shown several aspects of hinting in a learning context where the tutor acts as a helper for students to overcome an impasse. The human tutor provides more elicitative hints than didactic ones. The students tend to make opportunistic assertions that follow neither a strict forward chining nor a backward chaining. Accepting their reasoning style would be beneficial for students, but it makes the tutoring system more elaborated so that it must provide an appropriate hint depending on the students’ reasoning. The fact that nearly all the hints provides by the human tutor are about a single proof step motivated us to categorize hints as a Cartesian products of focus and form of hints. This formalization encompassed the hint events observed in our study. That is, the human tutor’s hint strategy might be modeled with the formal language of hints proposed here.

We hypothesize that maximizing students’ exploration (i.e., allowing their opportunistic assertion) will encourage their learning. One of the experimental questions that must be addressed next is which brings better learning gain for intermediate students: (1) to restrict them to the model inference (MTT), or (2) to let them follow the opportunistic assertions? The current study provides some crucial steps toward this goal.

Reference

Anderson, J. R., Boyle, C. F., & Yost, G. (1985). *The geometry tutor.* Paper presented at the Proceedings of the International Joint Conference on Artificial Intelligence, Los Angeles.

Corbett, A. T., & Anderson, J. R. (1992). LISP Intelligent Tutoring System: Research in skill acquisition. In J. H. Larkin & R. W. Chabay (Eds.), *Computer assisted instruction and intelligent tutoring systems: Shared goals and complementary approaches* (pp. 73-109). Hillsdale, NJ: Erlbaum.

Dreyfus, T. (1999). Why Johnny Can't Prove (with Apologies to Morris Kline). *Educational Studies in Mathematics, 38*(1), 85-109.

Hoffer, A. (1981). Geometry Is More Than Proof. *Mathematics Teacher, 74*(1), 11-18.

Hume, G., Michael, J., Rovick, A., & Evens, M. (1996). Hinting as a tactic in one-on-one tutoring. *Journal of the Learning Sciences, 5*(1), 23-47.

Kalyuga, S., Chandler, P., Tuovinen, J., & Sweller, J. (2001). When problem solving is superior to studying worked examples. *Journal of Educational Psychology, 93*(3), 579-588.

Koedinger, K. R., & Anderson, J. R. (1993). Reifying implicit planning in geometry: Guidelines for model-based intelligent tutoring system design. In S. P. Lajoie & S. J. Derry (Eds.), *Computers as cognitive tools* (pp. 15-45). Hillsdale, NJ: Lawrence Erlbaum Associates.

Matsuda, N., & VanLehn, K. (2000). A reification of a strategy for geometry theorem proving, *Intelligent Tutoring Systems, Proceedings* (Vol. 1839, pp. 660-660).

Person, N. K., Graesser, A. C., Kreuz, R. J., & Pomeroy, V. (2001). Simulating Human Tutor Dialog Moves in AutoTutor. *International Journal of Artificial Intelligence in Education, 12*, 23-39.

Sweller, J., & Cooper, G. A. (1985). The use of worked examples as a substitute for problem solving in learning algebra. *Cognition and Instruction, 2*(1), 59-89.

VanLehn, K., Lynch, C., Taylor, L., Weinstein, A., Shelby, R., Schulze, K., Treacy, D., & Wintersgill, M. (2002). *Minimally invasive tutoring of complex physics problem solving.* Paper presented at the 6th International Conference on Intelligent Tutoring Systems.

Zhou, Y., Freedman, R., Glass, M., Michael, J. A., Rovick, A. A., & Evans, M. W. (1999). *What should the tutor do when the student cannot answer a question?* Paper presented at the Florida Artificial Intelligence Symposium, Orlando, FL.