THE GEOMETRY TUTOR

John R. Anderson, C. Franklin Boyle, Gregg Yost

Advanced Computer Tutoring Project
Carnegie-Mellon University
Pittsburgh, PA 15213

Abstract

The tutor for doing proofs in high school geometry consists of a cot of ideal and buggy rules (IRR), a tutor, and an interface. The IBR is responsible for ehiuertly computing matcher, to all the correct and incorrect rules The interface is responsible for interacting with the student and graphically representing the proof. The tutor is responsible for directing the IBR and interface to achieve a current tutorial strategy. The strategy wo employ involves tracing the student's behavior in terms of what rules in the IBR it instantiates, correcting the student when behavior deviates below a minimum threshold, and helping the student over hurdles. While incomplete, current evidence indicates the geometry tutor is guite effective.

The Advanced Computer Tutoring Project has been working on the development of intelligent computerhased tutors for mathematics and science subjects in the range of senior high-school to junior college. This paper describes the general framework that we have developed and its instantiation in the case of a tutor tor generating proofs in geometry. First, we will describe the general philosophy of out tutoring efforts. Second, we will describe the basic structure of the geometry tutor that we have built. Third, we will describe the ongoing efforts to evaluate the tutor.

General Philosophy

Our approach to tutoring stems from two observations about the nature of learning in domains like high-school geometry. First, the major learning hurdle is acquiring domain-specific problem-solving skills (e.g, generating proofs in geometry), and the way they are learned is by practicing the problem-solving skill. We have intensively observed numerous students going through the first half of high school geometry (Anderson, 1001). We have not yet seen a case where a student was tible to absorb abstract instruction (from the text or the teacher) and go off and use that knowledge effectively in problem-solving. Students only acquired effective use of problem-solving knowledge by struggling with its application through a series of problems.

The second observation is that students learn these problem-solving skills much more effectively if they learn with a full-time, private, human tutor who knows the domain well than if they learn in the standard classroom situation, where students solve problems on their own, getting feedback by handing in exercises to be graded or comparing their solutions with example solutions In our work comparing these two modes of study, we have found the students with private tutoring reach the same achievement level as much as four times more rapidly than the classroom students. Bloom (1984) has compared private tutoring with classroom instruction for two school topics: cartography and probability. In his research, students spent the same amount of time learning, and he looked for differences in achievement levels. He found that 98 percent of the students with the private tutors performed better than the average classroom-student. Interestingly, he also found that the strongest benefit was for the poorest students. There was little difference between the achievement levels of the best students in the two conditions.

There are a number og reasons for the success of the private human tutor. Very important is the constant monitoring and shuctunng of the student's problem solving attempts The tutor can diagnose what contusions that paiticular student has, determine what the student needs to know, and provide instruction specific to these needs. the teacher can communicate the overall goal structure that controls the pioblem solving something that is difficult to do outside of the problem context and which is seldom attempted in classroom instruction. Private tutors are very good at managing student errors They provide immediate or near immediate feedback on these errors They can point the student back to the right track and prevent the student from getting lost. In this way students can partially solve problems that they would normally give up on and so can learn from their solution attempts.

The ability to be an effective tutor requires that the tutor have an internal model of how the skill should be performed and be able to access this model for purposes of instruction. Such a model we call an ideal model. The tutor needs to be able to supplement this ideal model with various errors that a student makes in deviating from the model. We call this the buggy model after Brown and Burton (1978).

One of the major commitments in our tutoring efforts has been to instantiate the ideal model and the buggy model as production systems. This follows from our psychological analysis of such skills in terms of the ACT* production system (Anderson,

1963). There are two major consequences of this commitment. First, ACT* production systems have well-defined goal structures which we try to communicate to the student. Second, these production systems imply a commitment to a grain size of instruction. Essentially, every production in the system encodes a meaningful step of cognition. Therefore, we monitor the student's problem-solving production step by production step and after each step determine which production (from the ideal or buggy model) that step instantiates.

This leads to what we have called the model-tracing paradigm for instruction. Essentially, the tutor traces the student's behavior through its ideal and buggy model. At any point in time there are a number of productions in the tutor that might apply. The tutor infers which rule the student executed by determining which one matched the student's output. If it is a correct production, the tutor slays quiet and continues to trace the students problem solving. If an incorrect production has bean applied, the tutor interrupts with appropriate remedial instruction The final possibilities are that the student does not Know what to do next or that the students behavior matches no production, correct or incorrect Usually, this occurs when the student is greatly confused We have found that the best thing to do in such situations is to tell the student the next step to take. If this is explained properly, the student is often able to get back on a right track.

The set of ideal and buggy rules (henceforth called IBRs) are based on considerable theoretical analysis of the characteristics of the problem domain and on a great deal of empirical observation of student behavior. Therefore, it is reasonable to suppose that they are more sophisticated than those possessed by most private tutors. Also, we have greater access to the knowledge encoded in these rules than do typical private tutors for whom the ideal model is often tacit and incapable of being directly described to students. We think it is possible to outperform human tutors in our use of these rules, which is not to deny that our computer tutors will be outperformed by human tutors on other dimensions such as natural language analysis.

The Architecture of the Geometry Tutor

In trying to achieve our tutoring paradigm, we have realized the need to separate out three components in a tutor. These are the IBR, the tutor, and the interface. These three components are clearly separated in the current version of the geometry tutor, although they were not so clearly separated in earlier versions (Boyle & Anderson, 1984). There are a number of advantages that come from clearly asperating the components. First, from a software engineering point of view it becomes possible to develop and test each component independently. Second, the components can be optimized for their distinct characteristics. Third this is a step towards one of the ultimate goals of the Advanced Tutoring Project--which is to develop a domain-free theory of instruction. While our current tutor is not entirely domain-free it represents a strong step in that direction. The IBR and the interface are the components in which most of the domain specifics are represented.

The centra! component in this triad is the tutor It defines our current tutoring strategy In our current system the tutor is implemented as a production system in the OPS5 language (Forgy, 1981). It uses information both from the IBR and from the student (via the interface) to control the tutorial interaction. It can present information to the student and lequest information from the student by invoking the interface It has two means of interaction with the IBR:

(a) It can look at which ideal and buggy rules are currently instantiated in the IBR and use these to interpret the students behavior. It allows the IBR to trace the student's solution by selecting in conflict resolution one of the IBR production rules. It then receives an update of the instantiated rules in the IBR after firing this rule.

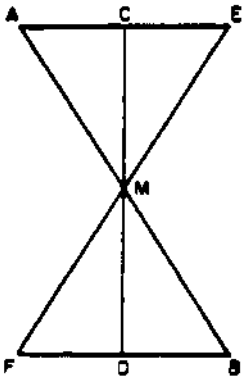
(b) It can request of the ideal model whether a statement can be proven, subject to certain constraints. This will cause the IBR to attempt a proof and report back information such as whether such a proof exists, how long it is, how optimal it is, what rules it involves, etc.

We will return to a further description of the tutor after describing the IBR and the interface.

The IBR

There were two major goals in our design of the IBR. The first was to have the ideal model generate proofs in what we felt was a natural, humanlike way--in contrast to some methods of proof. The bugs are hung off the structure generated by the ideal model so it is doubly important that it generate human like proofs, both in order to instruct students about what is right and to interpret their errors. The second goal was to make the IBR efficient. It is an important constraint on tutoring that the IBR perform its computations rapidly so that feedback to the student be rapid. Most of the computation in the geometry tutor is done in the IBR. As we will see, both of these goals posed interesting challenges in the case of geometry.

Figure 1 illustrates a geometry proof problem that is considered relatively complex for high-school students. The important feature to note is that at any point in time there are a large number of inferences that can be made. For instance, from the given fact that M is the midpoint of $\overline{AB}$, it is possible to infer that $\overline{AM} \cong \overline{MB}$. But this is just one of the many inferences that are possible. For instance, it is also possible to infer that $\angle AMF \cong \angle BME$ because of the vertical angle configuration formed by segments $\overline{AMB}$ and $\overline{FME}$. Moreover, these possible inferences can be *ordered* in a continuum according to aptness, where the first is very apt and the second is very inapt.



GIVEN M is the midpoint of $\overline{AB}$ and $\overline{CD}$

PROVE M is the midpoint of $\overline{EF}$

Figure 1

There is another kind of deductive reasoning that students engage in when faced with such problems This is reasoning backwards from statements to be proven to statements that will prove them Thus, a student can reason backwards from the goal of proving M is the midpoint of $\overline{EF}$ to the subgoal of proving $\overline{ME} \cong \overline{MF}$. This involves the definition of midpoint. It is possible to reason backwards from this goal again. For instance, one might reason backwards from the goal of proving $\overline{ME} \cong \overline{MF}$ to the subgoal of proving $\triangle AMC \cong \triangle BMD$ This would involve the use of the corresponding parts rule for congruent triangles. On the other hand, one might reason back from the goal of proving $\overline{ME} \cong \overline{MF}$ to the subgoals of proving $\overline{ME} \cong \overline{AM}$ and $\overline{AM} \cong \overline{MF}$. This would involve using the transitive property of congruence. Again, these backward inferences can be ordered as to their aptness with the first two inferences being quite apt but not the third.

The important observation is that the aptness of an inference is not an absolute property of the rule of geometry that authorizes it, but rather a function of the context in which it occurs. For instance, it does not seem appropriate to make the vertical-angle forward inference that $\angle AMF \cong \angle BME$. However, another vertical angle inference, $\angle AMC \cong \angle BMD$ is quite apt, particularly after we establish that $\overline{AM} \cong \overline{BM}$ and $\overline{MC} \cong \overline{MD}$. Then we can use these two side congruences and the angle congruence to show that $\triangle AMC \cong \triangle BMD$ by the side-angle-side postulate.

We have created an ideal model for generating proofs in geometry that involves forward and backward inference rules with contextual restrictions. It takes the form of a production system that enables us to model the flexible alternation between backward and forward reasoning that we observe in human experts, Every significant conlextualized rule of inference is a separate production rule Our use of a production system enables us to achieve the desired flexibility and to have the individual productions used as objects of instruction. Compared to other theorem provers for geometry such as Nevms (1074) our system is distinguished by the flexibility of its control structure and its decomposition of domain knowledge into relatively modular rules.

As an example of a forward inference rule, the following "Enflishified" production generates a vertical angle inference when that will enable a side angle side inference.

IF $\overline{XY} \cong \overline{UY}$ and $\overline{YZ} \cong \overline{YW}$

and there are triangles $\triangle XYZ$ and $\triangle UYW$

and $\overline{XYW}$ and $\overline{UYZ}$ are colinear

THEN infer $\angle XYZ \cong \angle UYW$ by vertical angles.

As an instance of a contextually bound backward rule, consider t

IF the goal is to prove two lines parallel

and there is a transversal

THEN set as a subgoal to prove the alternate

interior angles are congruent.

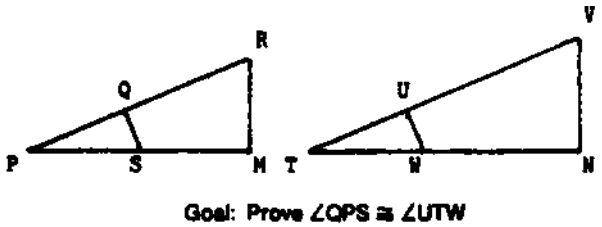
We have developed 300 such rules and have ordered them according to aptness. The ideal model applies the best inference rule that is satisfied in a situation, whether that is a backward or a forward rule. This system can prove all the problems in the high school material we have been working with. It also generates human-like proofs. Not all the inferences it makes are part of the final proof, but when it deviates from the final proof, it deviates in the way we have observed in human subjects.

Geometry poses an interesting challenge for efficient production-system implementation because of all of the symmetries that exist in geometry. Consider matching the rule that says if there is a goal to prove two ingles congruent and they are parts of corresponding triangles, set a subgoal to prove the two triangles congruent Below is the encoding that is close to what we actually use for this rule.

IF goal: $\angle abc \cong \angle def$
tests: there is $\triangle abc$
there is $\triangle def$
THEN goal: $\triangle abc \cong \triangle def$

Figure 2 illustrates a situation where this rule should match with two distinct bindings of variables (a = Q, b = P, c = S, d = U, e = T, f = W) and (a = Q, b = P, c = S, d = W, e = T, f = U). However, as the astute reader no doubt discerns, there is potential for a great many other bindings and the trick is to filter them out as unnecessary. Consider, for instance, the binding of a. Because of the symmetry of congruence and the symmetry of the end points of an angle, it can bind to any of P, R, S, M, U, V, W, or N. Having settled on this binding, there can still be two bindings for c, four for d, and two for f. (These four bindings completely constrain b and e.) Altogether there are 8 x 2 x 4 x 2 ■ 128 possible bindings to be considered, of which only eight

correspond to distinct hypotheses about triangle congruence. Six of these distinct hypotheses (e.g., $\Delta PRM \cong \Delta TVN$) can be rejected as implausible given the physical diagram-this plausibility check is the same idea used by Gelernter (1963). The only two plausible hypotheses about triangle congruence are $\Delta QPS \cong \Delta UTW$ and $\Delta QPS \cong \Delta WTU$.



Goal: Prove $\angle QPS \cong \angle UTW$

Figure 2

An immediate reaction to this problem is to try to canonicalize the angles, segments, and triangles in the diagram and have a unique name for each such object. The problem with such attempts is that they fail to uncover all the possible matches to geometry rules that exist. These can only be uncovered by considering the actual ordering of points as significant. For instance, most canonicalizations of this problem would fail to be able to retrieve the eight distinct hypotheses about triangle congruence in Figure 2. They would have particular difficulty with representing the distinction between the two plausible hypotheses above which only differ in terms of the correspondence between the points of the triangles. It is no accident that traditional geometry uses a point representation and not a canonicalized object representation.

Part of our solution is to encode into our statements of the rules the symmetries and have our pattern matcher take advantage of this. So for the rule above we inform the pattern matcher that it need not consider both angle orderings in the congruence. Therefore it can assume $\angle ABC$ will match to $\angle QPS$ and not worry about the potential match to $\angle UTW$. It also need not consider both orders of the ray points (A. C) in the first angle. Therefore it can assume A will match to Q. This reduces by a factor of four the amount of pattern matching that must be done. Secondly, we have a special coding for the points on a ray such that we do not have to consider each combination separately. For instance in Figure 2, two ray points can match to each of points A, C, D, and F in the pattern. However, we do not have to consider the $2^4 = 16$ combinations. So, rather than having 128 instantiations of the pattern $\angle ABC \cong \angle DEF$, we have 128/(4 x 16) - 2 instantiations. As most of the computation in the IBR is spent in pattern matching, this is a very significant savings. These two instantiations are expanded to eight when we consider the various triangles that can be formed, but six of these are filtered out on the basis of plausibility.

These patterns are organized on a modified Rete net to obtain much of the efficiency of the OPS family of production systems (Forgy, 1982). We estimate that we are able to obtain a couple orders of magnitude of efficiency over any previous production-rule-based geometry theorem prover. For instance, it is able to produce the prototypical expert solution to the problem in Figure 1 in eight seconds of CPU time on a Xerox Dandelion. Table 1 summarizes the sequence of inferences that it makes. It makes three inferences that are not part of the final proof, but these are inferences experts frequently make, and these inferences actually lead to a slightly longer proof. The important fact to note is that eight seconds is much less than human performance (usually a minute) and so is definitely within the bounds for acceptable tutoring.

Table 1

Inferences Made in Generating a Proof for the Problem in Figure 1

Backward Goal or Forward Inference		Statement	Rule	Time
Goal		1 $\overline{EM} \cong \overline{MF}$	Def-Midpoint-Backwards	17 sec
Inference		2 $\overline{CM} \cong \overline{MD}$	Def Midpoint-Forward	17 sec.
Inference		3. $\overline{AM} \cong \overline{MB}$	Def-Midpoint-Forward	22 sec
Inference		4 $\angle BMD \cong \angle AMC$	Vertical-Angles-in-Triangles-Forward	37 sec
Inference		5. $\Delta CMA \cong \Delta DMB$	Side-Angle-Side-Forward	15 sec
Inference		6 $\angle BMF \cong \angle AME$	Vertical-Angles-Known-Side-Forward	38 sec
Inference	*	7 $\angle DMF \cong \angle CME$	Vertical-Angles-Known-Side-Forward	53 sec
Goal	*	8. $\Delta EMC \cong \Delta MFD$	Corresponding-Parts-Segment-Backward	.38 sec
Goal		9 $\Delta EMA \cong \Delta FMD$	Corresponding-Parts-Segment Backward	53 sec
Goal	*	10. $\angle MEA \cong \angle MBF$	Angle-Angle-Side-Backward	.50 sec
Goal		11 $\angle MAE \cong \angle MFB$	Angle-Angle-Side-Backward	35 sec
Inference		12 $\angle MAE \cong \angle MFB$	Corresponding-Parts-Angle-Forward	4 27 sec

* denotes statement not part of final proof

The Interface

A major effort in the design of the interface has been to communicate to the student the logical structure of a proof and the structure of the problem solving process by which a proof is generated. Figures 3-5 illustrate the proof graph that we have developed for this purpose. There we have a representation of the graph at the beginning of a geometry proof, in the middle of the proof, and at the end of the proof. Figure 3 illustrates the initial problem-state. The statement to be proven is at the top of the screen, the givens at the bottom, and the diagram in the upper lefthand corner. The student can reason forward from the givens and backward from the statement to be proven. The student grows the graph by a combination of pointing to statements on the screen and typing in information. Each step of inference involves a set of premises, a reason, and a conclusion. Reasoning forward, the student points to the premises, types in the reason, and points to the conclusion or types it in. Reasoning

backwards, the student points to the conclusion, types in the reason, and then provides the premises.

Figures 4 and 5 show some of the possible states in proof development. The student is finished when there is a set of

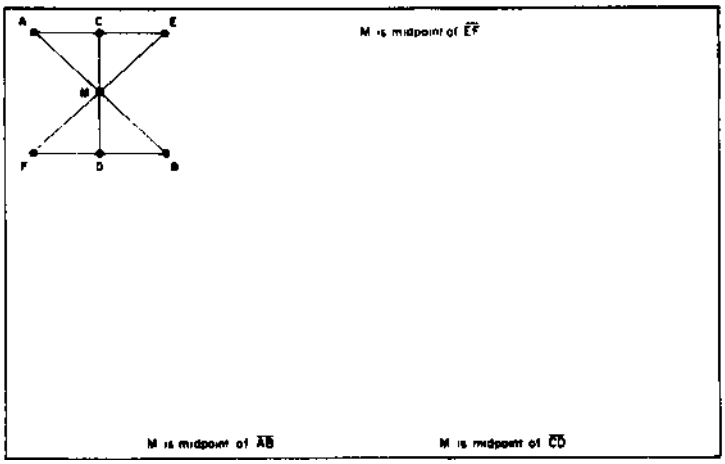


Figure 3 Initial Problem State

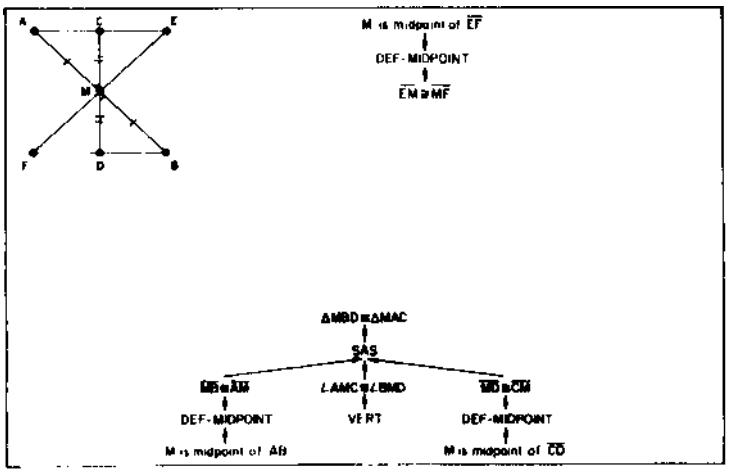


Figure 4 Immediate Problem State

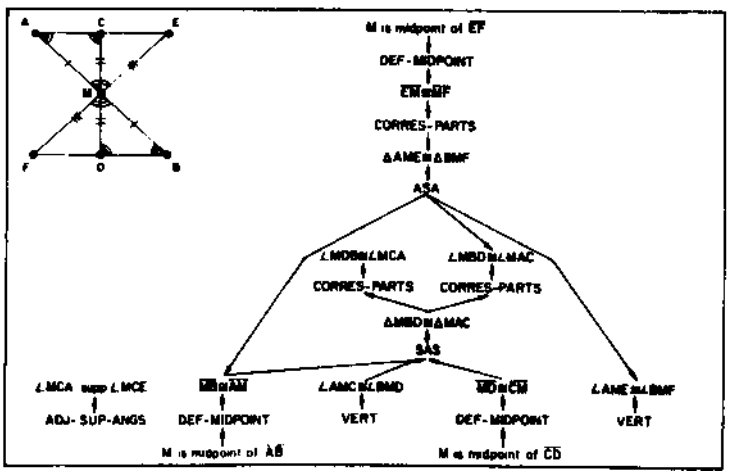


Figure 5 Final Problem State

logical inferences connecting the givens to the statements to be proven. Figure 4 illustrates how inferences can be grown from the top and the bottom to meet in the middle. Figure 5 shows the screen when a student achieves a final proof. Note that the student has made some inferences which were not part of the final proof.

One function of this formalism is to graphically illustrate the structure of a complete proof. High school students typically do not appreciate how the steps of a proof fit together and find this structure to be particularly enlightening. Second, the proof graph concretely illustrates critical features of the problem space-that inferences can be grown in forward and backward mode, that choice points exist where the student must choose among multiple inference rules, that the ultimate goal is to complete a well-formed logical structure.

There are a number of other features available in the interface which the student finds uselul. The system automatically does spelling and syntax checking and so protects the student from the misundurstandings that can result from mistyping. Numerous help windows can be brought up. The two help windows that students find most useful are a window providing the currently applicable rules of inference and a window giving the statement of any particular postulate, theorem, or definition.

The Tutor

The tutorial component can be described at two levels. First, there is the minimal tutor that we have actually implemented and tested with students, and then there are the various embellishments that we are in the process of implementing and testing. The minimal tutor can be described with respect to three steps a student must go through to complete an inference: selecting a set of statements from which to make an inference, specifying the rule of inference that will apply to these statements, and then specifying the statements that result from applying this rule of inference.

In the selection step the student can either select a set of statements from which to apply an inference rule or ask for help. If the student asks for help, the tutor picks the statement set of the most highly rated rule in the conflict set of the IBR, subject to the constraint that the inference produced by that rule is actually part of a proof. It provides the student with the statements (backward or forward) that are part of this rule. If the student selects a set of statements and they represent the instantiation of any legal rule, the system accepts it even if it is poorly rated and even if it is not part of the proof. If the student selects a statement set that is not part of any rule, the tutor tells the student so, and the student gets to select again. If the student chooses a wrong set again, the tutor presents the student with the same statement set that It would if the student had asked for help. This "two strikes and you're out" principle was instituted to prevent the student from looping with guess after guess.

The tutor treats the rule-selection step similarly. If the student asks for help, it provides the highest-rated rule involving the statements chosen in the previous step. It will accept any legal rule that applies to those statements. If the student twice provides illegal rules, it will give the highest-rated legal rule. The analogous thing is again done in specifying statements that are the result of applying the rule.

This minimal tutor is one that tries to guide the student to a solution with minimal intervention It accepts any legal inference that the student wants to make We have found that students can get lost in a maze of legal but useless inferences. Therefore, one thing we want to do is to prevent students from making low-rated inferences that are not part of any reasonable proof. The minimal feedback we can provide would be something like "It is legal to do that, but it won't get you anywhere." However, we would like to build into the tutor recognizers for particular false paths as well as unnecessarily long-winded paths that students go down. As an example of the latter, students often prove theorems as part of a larger proof when they could apply that theorem directly.

Implied in this capacity is the ability of the tutor to query the IBR as to whether an unanticipated inference is part of a reasonable proof. In reasoning backwards this amounts to judging if each of the premises can be proven and how quickly. In reasoning forward this amounts to judging whether the conclusion of a forward inference figures in a reasonable proof.

With these facilities in place we can begin to explore the issue of how directive to be in our feedback. We can allow students to go a fixed number of steps off the path, to override the tutor's advice, etc. Given our commitment to immediate feedback and reducing working-memory load, we would expect that the more structured environment would be better, but there is sufficient controversy on this topic to merit empirical exploration.

Another feature we would like to augment the tutor with is the ability to recognize student confusions and instruct on these confusions. For instance, many students do not realize that in the side-angle-side postulate the angle must be included by the two sides. This requires that there be a buggy rule in the IBR to match this pattern and that the tutor know of this bug and have an explanation associated with it.

A related feature is the ability to give the student strategic advice in addition to simply telling the student that an inference is inadvisable. For instance, if the student starts out in Figure 1 with the inference $\angle AMF \cong \angle BME$, we would like to be able to say something of the order "No, it is not useful to make that vertical angle inference here. It is useful to make the vertical angle inference when the angles are corresponding parts of triangles you want to prove congruent. In this problem why don't you try to make an inference involving the fact that M is the midpoint of $\overline{AB}$."

Yet another feature we would like to provide the tutor with is the ability to generate remedial problems tailored to student weaknesses. This requires keeping an assessment of how well the student knows the various rules in the ideal model. If there is a weak rule, the tutor can generate a problem involving this rule and tutor the student on the rule.

Evaluation

Beginning in the fall of 1985 we intend to have a classroom of 10 Xerox Dandetigers in one of the Pittsburgh Public High Schools, and to teach four or five geometry classes of ten students each based on these tutors. The tutor will be devoted to the proof generation portion of high-school geometry, which is slightly more than half of a standard curriculum. This part of the course will be taught on a self-paced basis. The other part of the course (involving ruler and compass construction and coordinate geometry) will be taught in the conventional way. There will be many issues concerned with coordinating such a class which we have yet to work out. However, the outcome of this project will be a fairly systematic evaluation of the tutor and variations on It.

In absence of that we have only our pilot work with the geometry tutor to report. We have run three students through the minimal tutor in various stages of development. One student was of above-average ability, one of average ability, and one of below-average ability (as defined by their math grades). The below-average student came to us for remedial purposes, having failed 10th grade geometry. The other two were eighth graders with no formal geometry training. All learned geometry quite successfully and reached the point where they were solving problems more complex than are assigned in the Pittsburgh Public Schools. After it was over, all claimed to like geometry, which is encouraging given that classroom geometry is usually rated as the least liked of all school subjects (Hoffer, 1981).

Assuming that we can establish equally positive results on a larger scale, the next question concerns how we can reproduce this tutor on a more economical scale. It is our belief that the generation of personal computers slated for the late 1980s will be sufficiently powerful to deliver this instruction. For instance, the IBM machine being developed on the CMU campus is projected to be sufficiently powerful. Moreover, there are optimizations in the direction of compiling the real time computations made by the IBR and the tutor that might reduce computational time by a factor of ten.

References

Anderson, J.R. Tuning of March of the problem space for geometry proofs. Proceedings of IJCAI.81.1981, pages 165-170.

Anderson, J.R. The Architecture of Cognition. Harvard University Press, Cambridge, MA, 1983.

Bloom, B.S. The 2 Sigma Problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher 13:3.16. 1984.

Boyle, C.F. & Anderson, J.R. Acquisition and automated Instruction of geometry proof skills. Paper presented at the 1984 AERA meetings, 1964.

Brown, J.S. & Burton, R.R. Diagnostic models for procedure! bugs In basic mathematical skills. Coonitive Science 155.192 1978.

Forgy, C. L OPS5 Manual. Carnegie-Mellon University, Computer Science Department, 1981.

Forgy, C.L Rett: A fast algorithm for the many pattern /many object pattern match problem. Artificial intelligence. 19:17.37 1982.

Qelernter, H. Realization of a geometry theorem-proving machine. *intelligence* Tutoring Systems McGraw-Hill, New York, 1963.

Hoffer, A. Geometry is more than proof. Mathamatics Teacher : 11 January, 1961.

Nevins, A. J. Plane geometry theorem proving using forward chaining. Al Memo 303. Cambridge, MA: MIT Al Laboratory, 1974.