Term Project Paper

85-411: Cognitive Processes and Problem Solving

A Strategy to Prove Geometry Theorems with Construction:

--- Could they say yes, or no? ---

Noboru Matsuda
Intelligent Systems Program
University of Pittsburgh

April 30, 2001

Abstract

Geometry theorem proving is a challenging task, especially when it involves construction – namely, to add new segments and points to a problem figure to complete a proof. Little is known about expert performance on construction. There exists a model of theorem proving with construction as a state-space search, but appears to be computationally beyond the human reasoning capacity. It is also known that for geometry theorem proving, the ratio of the number of irrelevant paths to the number of solution paths is very high. Hence, it might be more efficient to use heuristics to select only appropriate paths instead of rejecting inappropriate ones. However, our experiment showed that an experienced subject in theorem proving preferred to use rejection heuristics. This paper examines search heuristics that human provers can use to explain this unexpected observation. Our ultimate goal is to build a cognitive model of theorem proving with construction for a computerized tutoring system that can help students to construct such proofs by themselves. In this paper, we also discuss several factors that make constructions so difficult.

1. Introduction

It is a hard task both for human and automated problem solvers to prove Euclidean geometry theorems – especially when it involves constructions. Construction is the activity to draw additional points and segments that are not appeared in an original problem figure but are necessary to complete a proof. A pair of compasses and straightedge is only the tool that can be used for construction. One can assume to move the spike of compass to draw several arcs with the same radius (i.e., the compass isn't necessarily *collapsible* as Euclid originally assumed). However, the straightedge must be *unruled*, namely, one can't transfer segments with a straightedge[1]. Hence, there are only three primitive operations: (a) to draw a straight line with a straightedge, (b) to apart the compass by the distance in between two existing points, and (c) to draw an arc with a compass.

Fig. 1 is an example of geometry theorem that requires construction. This theorem is proven, for example, by drawing segment CQ and extending it to segment AB. Let X be the intersection of line CQ and AB. With this construction, one can conclude that CQ = QX (for ΔDQC ≡ ΔBQX) hence PQ // AB that in turn concludes DM = MA (by the midpoint law for ΔDAB).

Fig. 1: An example of theorem that requires construction.

Indeed, theorem proving with construction is one of the most difficult tasks and it has yet been fully understood what makes the task so difficult in any discipline. Very little is known about experts' knowledge of construction in any field including education (Polya, 1957), artificial intelligence (Matsuda and VanLehn, 2001), and cognitive science (Itho, Ohnishi, and Sugie, 1993).

Without a full insight, theorem proving with construction has been taught in schools in many countries. Japan, China, and France are the most active countries where they spend for hours to teach theorem proving. In those countries, both teachers and researchers believe that construction is a creative, fascinating, and strategic activity, hence deserves to be a subject in elementary geometry. Indeed, the theorems that require construction are used in entrance examinations for high school in Japan. However, the teachers' ignorance of the subject itself (i.e., the lack of expertise of construction) makes students

[1] Consider a construction shown below that is prohibited by unruled straightedge: Given two segments L and M, draw a segment with the distance *d* starting from a point on L and ending at a point on M while passing through a point X, which is neither on L nor M. This construction is possible only with a ruled straightedge with which you can measure *d*.

difficult to learn theorem proving with construction. It is critical, at least for educational purpose, to understand how human experts build up a proof with constructions.

Theorem proving with construction can be formalized as a well-defined problem. Indeed, it is possible to formulate the task with finite predicates and operators. Thus, according to the theory of general problem solving (Newell and Simon, 1972), we should be able to formulate a problem space and search for the proofs with construction. However, recent research shows that a search for construction is computationally very complex (Matsuda and VanLehn, 2001) and it is unlikely that human problem solvers could maintain such search. Nonetheless, human being can find a proof with construction rather efficiently. Why?

This study is to investigate how human problem solvers find construction with less effort. Specifically, we are interested in extracting knowledge to control search that the human problem solvers can use. Basically there are two heuristics to minimize the search: (a) to *select* only the paths that lead to a solution, (b) to *reject* the paths that don't lead to a solution. If a theorem prover has either technique with the 100% accuracy, then those two techniques are identical. However, it is unlikely that human prover could have such an accurate search technique. Hence, it would be hypothesized that the human provers only use either technique, or both, with a limited accuracy. This hypothesis further predicts that, with regard to appropriateness of an inference that they are about to draw, human experts must be able to test the appropriateness of their inference by saying "yes" (i.e., select the inference) or "no" (i.e., reject the inference). If human experts apply selection heuristic, then by extracting such heuristic from human experts, we should be able to teach students how to see the correct paths to follow. If it is rejection heuristic that human experts apply, then we should be able to teach students how to smell out flaws of their inferences.

This paper examines the heuristics for search, which human problem solvers can use, to build a cognitive model of theorem proving with construction. To analyze the process of problem solving, we apply verbal analysis (Chi, 1997) that is suitable to captures the representation of the knowledge involved in the problem solving. We first introduce a model of theorem proving with construction as a state-space search in section 2. Several implications from experimental study on theorem proving with construction are described, which in turn support the hypothesis on the search heuristics that the human prover could acquire. The hypothesis is that human provers must apply *selection* heuristic, namely, instead of pruning the irrelevant paths, the highly efficient prover should be able to select appropriate paths. We discuss this hypothesis in section 3. Section 4 explains the details of experiment followed by the results in section 5 where we found that the subject involved in our study mostly relied on *rejection* heuristic. Finally, in section 6, we discuss why the subject tended to apply rejection heuristic even she has enough experience in geometry theorem proving.

2. Theorem Proving with Construction as a State-Space Search

Geometry theorem proving can be modeled as a state-space search (Newell and Simon, 1972). A state consists of (1) a problem figure, (2) a set of propositions either given or derived, and (3) a goal to prove. The problem figure is essential in the state because the constructions are to change the configuration of problem figure.

There are two different types of operators: (a) the operators to logically deduce propositions and (b) the operators to construct new points and segments. The former is a set of axioms, definitions, and the theorems that have been proved. In this paper, we refer them to as the *postulates*. A postulate is in a form "If a set of *premises* holds, then a *conclusion* also holds." A postulate is called applicable when all premises are held. Once a postulate is applied, the conclusion is asserted as a new proposition.

The primitive operators for construction are (1) to draw a line with straightedge, (2) to open a pair of compasses to a specific distance determined by two existing points, and (3) to draw an arc with the compasses opened to a specific or arbitrary distance.

With those operators, a straightforward implementation of the search for theorem proving with construction would then be as follows:

1. Repeat to apply applicable postulates until no new propositions are asserted.
2. If the goal unifies one of the propositions asserted, then we are done (i.e., a proof has been found as a history of postulate application).
3. Otherwise apply any of the construction operator(s).
4. Repeat from the step 1.

To understand the step 1, we need to clarify the extent to which the above algorithm works. In this study, we only deal with theorems that don't involve any algebraic operations (i.e., sums). This restriction is necessary to ensure that the search strategy described here would work. In other words, if we include algebraic operations, then the exhaustive application of postulates at step 1 might be infinite.

Above search procedure is complete, namely, if a proof exists, then the search procedure will eventually find it, because the construction used in the proof must be eventually find by the procedure at step 3.

Unfortunately, however, the search complexity is dominated by the complexity of finding an appropriate construction at step 3. Therefore, this procedure is not tractable for the theorems with complex constructions.

One way to make above procedure tractable is to use macro operators for construction. Matsuda and VanLehn (2001) built a theorem prover, called GRAMY, that is an implementation of improved version of above search procedure. The basis idea for construction is to extend the problem figure so that a postulate with a conclusion that unifies the goal to be proven would apply (such postulates are referred to as *useful postulates*). Each postulate in GRAMY is codified with a geometric configuration associated. The construction technique used in GRAMY is as follows: select a useful postulate, calculate a partial match between a configuration of the useful postulates and a configuration of the problem figure, and make up the gap. GRAMY uses a set of macro operators to make up the gap for the final step. Notice that construction is achieved by a backward application of the useful postulates.

Matsuda and VanLehn showed that GRAMY actually solved most of the difficult theorems. Their findings include the following:

- Most of the theorems have a proof with a single construction that can be made at the initial state.
- Some theorems require multiple constructions, i.e., multiple backward applications of the postulates that result in different constructions. And they are computationally very complex. However, those problems are rare in the literature.

3. How could human problem solvers control their search?

Although GRAMY's search algorithm is computationally tractable, it is unlikely that that human problem solves can follow the same procedure. Even without construction, the search might blow up. Table 1 shows the search complexity for the theorems that have a proof with single construction at the initial state. Construction technique used in GRAMY would not always be tractable for human problem solver, since there might be too many possible constructions.

Table 1: Search complexity the theorems without construction.

Nonetheless, expert human problem solvers can prove all the problems quite efficiently. Therefore, they must have highly efficient search heuristics. Basically, there are two types of heuristics: (1) to select only appropriate operators among equally applicable ones, or (2) to reject inappropriate operators. If the heuristics were 100% accurate in either way, then they would equally work for both types (i.e., selection and rejection are identical). However, if the prover applies selection heuristics that have a flaw, then completeness of the search wouldn't hold (i.e., some proofs might not be found). If the prover applies rejection heuristics that have a flaw, then soundless of the search wouldn't hold (i.e., wrong proofs might be made).

It is unlikely that human provers could have perfect heuristics. Therefore, it would be hypothesized that we rely both or either on rejection or selection heuristics. In general, for geometry theorem proving, the ratio the number of nodes in correct paths to the number of nodes in entire search space is fairly small. For construction, for example, it is about 25%. That is, only about one fourth of plausible constructions lead correct proofs. Hence, an efficient prover must apply selection heuristics; they must *know* which operator should apply.

To see how human provers actually control their search, an experiment was planed and the protocol data were analyzed as explained in the next section.

4. Experiment

There has been no cognitive model for theorem proving with construction to represent expert knowledge. Therefore, it is difficult to predict the experts' performance on search control. To capture the heuristics that the expert provers use, the experts' inference processes were recorded and analyzed. This section describes the experiment where a subject was asked to prove a theorem with construction.

4.1 Method

A subject was asked to think aloud while proving the theorem shown in Fig. 1. There are 25 constructions created by the construction technique mentioned in section 2. Out of those, 4 constructions successfully lead proofs. The shortest proof for this theorem consists of 1 construction (at the initial state) and 7 postulate applications that follow the construction. There are 115 different propositions that can be derived for the shortest proof. A backward search expands 8.86 million nodes with the average branching factor of 6.99. Obviously, this kind of search is not tractable for human prover without using efficient heuristics.

There was 1 Japanese graduate student (referred to as CK hereafter) participated in the experiment. Instead of using several subjects and drawing statistical inferences, we used only 1 subject to apply qualitative analysis thoroughly to the protocol data. As an average Japanese graduate student, she had enough knowledge of theorem proving without construction. As shown later, this is supported by the fact that she could immediately find a proof when a correct construction was provided as a hint.

Before the tape-recorded session started, a cryptarithmetic problem, which is not related to geometry, was provided as a worm up puzzle to practice to think aloud.

The theorem was given as a printed material and the subject were allowed to use pencil to draw the lines and the marks on the problem figure. The subject was informed that construction is necessary to prove the theorem.

The subject was free to access a note that has a list of geometric postulates sufficient to prove the theorem given. The postulates are roughly corresponds with those that are taught in a geometry class in middle schools. Since the main purpose of the experiment is on the search heuristics, the open-book experiment was planed to prevent the subject from getting stuck due to the inaccessibility to knowledge they actually "understood."

To elicit more verbal data, an instruction about the construction procedure was provided if the subject couldn't find an appropriate construction within 20 minutes.

4.2 Analysis

The protocol data were segmented and coded with the labels shown in Table 2. They are categorized into 5 classes.

(1) Reference to an element of the problem: The utterances that imply that the subject was focusing on either geometric object or geometric axioms.
(2) Forward chaining: The utterances that imply that the subject was attempting to assert a proposition that is deductively inferred from current situation.
(3) Backward chaining: The utterances that imply that the subject was attempting to make a backward inference.
(4) Attempt at construction: The utterances observed when the subject was attempting to make a construction.
(5) Heuristic: The utterances that imply that the subject was attempting to apply a heuristic for search control. This includes a rejection of an assertion mentioned earlier, that is, the utterance that indicates that the subject was not convinced of (or even incredulous about) a preceding inference.

Table 2: The labels used for the coding.

5. Result

This section shows quantitative and qualitative characteristics found in the protocol data analyzed in the previous section.

5.1 Overall performance

We start with inspecting how CK utilized the heuristics to maintain search. The numbers in Table 2 shows the frequencies each utterance was observed. CK made twice as many backward chaining as forward chaining. Backward chaining and heuristics account for 58% of CK's thought. Interestingly, out of 19 heuristic applications, 18, or 95% were to reject reasoning that CK didn't think it would be successful (they were scattered around the codes `Helpess`, `SoWhat`, and `NotActually`). That is, she tended to draw an inference without strong convince at first and then abandoned it later on. The only selection heuristic she applied, which is labeled as `Strategy`, was "*to use premises given*." This fact indicates that CK mostly relied on rejection heuristics, which prunes inappropriate search paths.

Nonetheless, the fact that she didn't exhaustively search through the entire problem space indicates that she implicitly applied some sort of filtering or selecting strategy to choose "appropriate" inference steps. How could she maintain such reasoning?

Table 3 shows the utterances in a chronological order from top to bottom in rows. They are arranged in the columns according to the coding labels. There were 18 line-of-reasoning observed as indicated by dotted lines. Two dotted lines from the same cell show that CK made a back up (in backward reasoning) for the same goal.

Table 3: Overall performance of CK solving problem #1.

There were 3 hints provided in this session. They were shown as the thick gray lines appeared at the end of Table 3. The first hint told of the construction technique in general terms, that is, the backward application of a useful postulate to find a partial match that in turn indicates a necessary construction. The second hint told that the midpoint law is useful for the present theorem. The last hint directly told a partial match for the useful postulate that makes the useful postulate be applicable, that is, it said (originally in Japanese): "*by drawing a segment from C to Q and extending it all the way down to AB, you would see that the midpoint law could apply*."

CK seemed not to understand the first hint. After the hint, she drew segment QA that is hardly related with any postulate. Indeed, she didn't mention any postulate at all. The second hint made CK seek a pattern in the problem figure that would partially match with the midpoint law. However, she failed to find the pattern. This means that simply knowing a useful proposition is not sufficient to find a construction. The capacity of perception for a person who is not familiar with the partial match is apparently not powerful enough to apply the construction technique used in GRAMY.

Interestingly, when the last hint was provided, CK immediately found a proof all by forward chaining (those data have been omitted from the Table 3). This implies that she had had all knowledge necessary to prove this theorem, but just couldn't "see" a pattern for construction hidden in the problem figure.

5.2 Pattern of reasoning

In Table 3, it can be seen that CK relied mostly on backward reasoning and some forward chaining scattering here and there. Did CK systematically develop the inference? Or, she took rather trial-and-error type of reasoning? To see the reasoning pattern involved in the CK's protocol, we extracted the lines of reasoning from Table 3 and lined them in Table 4.

Table 4: The lines of reasoning

In the table, each line shows a chain of consecutive applications of postulate. In a backward reasoning, an application of postulate is considered as *consecutive* when the postulate has a conclusion that unifies the goal mentioned immediately before its application. In a forward reasoning, a consecutive application of postulate is the one that has premises that unify the propositions asserted immediately before the postulate's application.

The columns in each line show the utterances (simplified) in a chronological order. A hatched rectangle on the gray thick line indicates a flaw of the reasoning, i.e., inappropriate or not logically valid. An utterance above the gray thick line indicates that it is made backwards. An utterance below the line indicates that it is made forwards. For example, "bisector" in the first column on the line #2 means that CK mentioned that it is sufficient to prove that M is a mid point of DA. In this case, the actual utterance (in Japanese) observed is something like "*so, I need to prove that this segment is divided into two equal segments*." The second column on the same line indicates that CK then further inferred that it is necessary to prove MQ//AB to prove that DA is bisected, and so on.

A vertical line shows a back up, that is, CK tried alternative way to achieve an ancestral goal. For example, the third line shows a remedial line of reasoning for the goal MQ//AB.

The CK's suspicion appears in the parentheses below or above the utterance that is suspected. There basically 3 different kind of suspicions.

(1) `Sowhat`: CK was not sure how the proposition just made contribute for the proof.
(2) `Helpless`: CK thought that the proposition just made wouldn't contribute for the proof.
(3) `NotActually`: CK thought that the proposition just made was not true.

A shaded utterance shows a construction where `Perpendicular(P,XY)` means a perpendicular line from P to XY, `Extend(XY)` means an extension of line XY, and `Seg(XY)` means a segment connecting X and Y.

There were several interesting inference pattern seen in Table 4. First, CK made a big chunk of reasoning at the beginning of the session (from the line #2 to #6). They were mostly backward reasoning: there were 32 utterances including 10 utterances for rejection heuristics. Out of remaining 22 utterances, 19, or 86% indicated backward reasoning. Although, the first two subgoals (i.e., bisector and MQ//AB) were correct, she failed to partially match midpoint law to find a construction.

The back-ups for alternative reasoning in this big chunk were made both to generalize and specify a goal. For example, the goal ∠DQM=∠CDQ on the line #4 is a specialized alternation for the goal to find equal segments on the line 2. On the other hand, the goal to find parallel segments on the line #6 is a generalization of MQ//AB on the line #2.

A series of forward chaining was observed after the big chunk of backward reasoning. They seem to be made randomly. In other words, CK appeared to draw forward inferences in rather an ad-hoc manner.

Next question is then about the heuristics of construction: How did CK carry out the constructions? Since the three hints provided in the session are all about construction, it would be interesting to focus on the 4 constructions that were made before the first hint. Notice that the hints were given after the lines #14, #16, and #18.

All of the constructions were made by backward reasoning to achieve a desired configuration. For example, the first construction is planed to make a right angle that was necessary (as CK believed) "*to make two similar triangles*" that could further contribute to prove MQ//AB. However, those constructions were not always logically sound. For example, the segment PB was drawn (line #6) "*to use the premise AP=PC*," but PB has nothing to do with AP=PC and indeed CK couldn't draw any inference with those segments (PB, AP, and PC). The construction in the line #2, i.e., the perpendicular line from R to AB was drawn "*to make a right angle*" that was supposed to support MQ//AB with some similar triangles. However, the constructed perpendicular line was not actually related to any similar triangles that support MQ//AB.

Those observation on CK's inappropriate reasoning motivated us to further investigate her skills of theorem proving, especially the mastery level of geometric postulates and their application. That is, how well could CK access to appropriate postulates and apply them in an appropriate way?

5.3 Application of Axioms

Table 5 shows the geometric postulates that were apparently used in CK's reasoning. The postulates are placed on the same skeleton used in Table 4. The question marks (?) show that no specific geometric postulate could be identified from the utterance.

Table 5: Pattern of postulate applications.

It appeared that the constructions made by CK seemed not to be based on geometric postulate in any logical sense. Rather they appear to be motivated to "visually" relate with the precedent utterance. For example, see the line #6 in Table 4. The second construction, a set of perpendicular lines, was intended to make "*something that is relating with the given premises*," which in this case is DQ=QB. As a result, the perpendicular segments made two triangles that include DQ and QB.

CK apparently understands the midpoint law. Indeed, there were 3 utterances hat explicitly mentioning the midpoint law. Nonetheless, she didn't realize that the midpoint law is useful for construction. Despite of her competence for, or her familiarity with the midpoint law, she couldn't even "see" its partial match in the problem figure.

6. Discussion and Concluding Remarks

Despite the fact that theoretically selection heuristics could work much better than rejection heuristics for geometry theorem proving, we observed that CK strongly relied on rejection heuristics. Since the subject CK has enough experience in geometry theorem proving at least for hours in her middle school mathematics classes, we assume that her preference on rejection heuristics had been learned from the examples she solved. Why rejection? One possible interpretation is that a part of search space that human prover can see might be very small and in that limited space the ratio of inappropriate states to appropriate ones might be low. Another view is that it is easier to see a flaw of inference than to convince the prover that he is on an appropriate path; single counter evidence can refute the inference, but dozens of features might not be enough to select a path to go. In either way, it might be interesting to see if it is possible to teach selection heuristics to the students.

Observed constructions were all based on backward inference. This is consistent with expert-novice differences; the novices tend to carry out many backward reasoning with back up, while the experts draw appropriate inference by forward reasoning. Other interesting observation is that the inferences involved in construction were not always logically sound and mainly motivated by visual cues. Presumably this is because the students learn the visual cues through out their experience on theorem proving (Zhu, Lee, and Simon, 1996). It has yet been fully investigated what aspect of the description of theorem could be a cue.

The hints about the construction technique didn't help the subject at all. Why the construction heuristics didn't work? There are two issues. First of all, it might be hard to aware the useful postulate. Indeed, the first hint mentioning to find a useful postulate didn't work. Second, it might be hard to see the partial match that makes the useful postulate applicable. In our experiment, the subject didn't recognize the she eventually draw correct line (the extension of CQ to AB). So, to be familiar with the construction technique mentioned in this paper, it would be necessary to learn associative knowledge between the pattern of goal to prove and the postulate that is useful for that goal, as well as the partial match that is essentially to read a part of the configuration associated to the postulate off the problem figure.

7. Reference

Chi, Michelen T. H. (1997). Quantifying Qualitative Analysis of Verbal Data: A Practical Guid," Journal of the Learning Science, 6(3), 271-315.

Itoh, T., Ohnishi, N., Sugie, N. (1992). An Experiment and A Model to Investigate How We Use Diagrams in Geometric Problem Solving, Proceedings of The 2nd Pacific Rim International conference on Artificial Intelligence, pp.1072-1078.

- They had further developed a model of human problem solving with diagrams. Unfortunately, they don't have any publications for the latest model, but do have some in Japanese. Itoh, T., Ohnishi, H., Sugie, N., (1994). Why do we feel easy to solve some problems by drawing diagrams? Transactions of Japan Society of Information Processing, 35(7), pp.1501-1505.

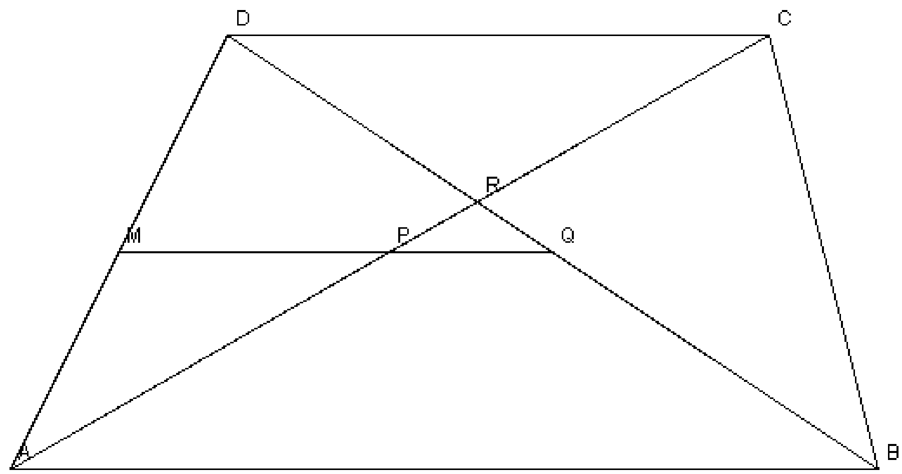
Matsuda, Noboru & VanLehn, Kurt. (2001). GRAMY: A theorem prover for geometry theorem proving with construction. Unpublished document (prepared for publication).

Newell, A., and Simon, H. A. (1972) Human Problem Solving. Prentice-Hall, Englewood Cliffs, NJ.

Polya, G. (1957). How to solve it. Princeton, NJ: Princeton University Press.

Zhu, Z., Lee, Y., & Simon, H. A., “Cue recognition and cue elaboration in learning from examples,” Proc. Natl. Acad. Sci. USA., Vol.93, pp.1346-1351, 1996.

Fig. 1: An example of theorem that requires construction.



Given: DQ = QB

AP = PC

DC // AB

Goal: AM = MD

Table 1: Search complexity the theorems without construction.

Problem	Proof Length	Forward Chaining					Backward Chaining				
		Search Depth	Time	Space			Search Depth	Time	Space		
				#Node	#Facts				#Node	ABF	#Facts
P007	2	1	0.83	3	11		1	0.83	3	8.67	3
P001	3	2	1.54	6	22		2	1.54	30	5.03	2
P002	4	2	5.27	7	50		3	5.44	229	5.42	2
P008	4	2	4.40	7	34		3	4.56	57	6.16	3
P010	6	4	9.40	15	57		5	8.84	7104	6.84	2
P006	7	2	29.55	6	146		6	31.03	360	2.26	4
P004	10	5	28.23	42	111		9	106.56	1267388	3.89	5
P011	40	12	41.19	2259	198		14+	-	40378455+	0.00	6
P005	55	10	37.02	498	199		9+	-	53324115+	5.93	7

Table 2: The labels used for the coding.

Reference		Forward chaining		Backward chaining		Construction		Heuristic	
READ-given	3	FC	8	Wish	15	Construction	9	Helpless	13
Focus	2	Seek-FC	4	Seek-pattern	6	Seek-Const	3	SoWhat	3
READ-goal	1	Assumption-FC	1	BC	4			NotActually	2
Seek-Axiom	1							Strategy	1
(8%)	6	(17%)	13	(33%)	25	(16%)	12	(25%)`	19

READ-given	Read givens specified in the problem.
READ-goal	Read goals specified in the problem.
Focus	Pay attention to a particular configuration in the problem figure.
Seek-Axiom	Refer to the notebook on geometric axioms.
FC	forward inference.
Seek-FC	Think of a logical consequence of a particular proposition(s).
Assumption-FC	Assume a proposition and think of a logical consequece from it.
Wish	Wonder if particular proposition would hold.
Seek-pattern	Seek a particular configuration in the problem figure that satisfies certain property (e.g., a triangle with "this" segment).
BC	backward inference.
Construction	Draw new segments.
Seek-Const	Seek if particular construction is possible.
Strategy	Apply a strategic search control.
Helpless	Dismiss a proposition that seems not to be useful.
SoWhat	Dismiss a proposition that leads no farther inference.
NotActually	Dismiss a proposition that seems not to be held.

Table 3: Overall performance of CK solving problem #1.

Reference				Forward Chaining			Backward Chaining			Construction		Heuristics			
4.11%	1.37%	2.74%	1.37%	10.96%	5.48%	1.37%	5.48%	6.85%	17.81%	4.11%	12.33%	5.48%	13.70%	5.48%	1.37%
3	1	2	1	8	4	1	4	5	13	3	9	4	10	4	1
READ-given	REF-goal	FOCUS	SEEK-axiom	FC	SEEK-fc	ASSUMPTION-fc	BC	SEEK-pattern	WISH	SEEK-const	Construction	sowhat	helpless	NotActually	STRATEGY
BQ=QB AP=PC DC//AB		1													
				Trapezoid											
	AM=MD														
							bisector MQ//AB (top-goal)		2						
									eq_angle eq_segment						
									MQ//AB						
		AMP													
									Similar Triangle						
									Construction						
										Construction					
			brawse												
									Construction Right Angle						
											Perpendicular (R, AB)				
									eq_angle ∠DMP=∠DAB			3			
														∠DMP=∠DAB	
												4			
									∠DMB=∠DMP or ∠DAB						
														∠DMB=∠DMP or ∠DAB	
							∠DQM=∠CDQ (MQ//AB)	5							
								Triangle with ∠DQM, ∠CDQ							
															Use Premise
							6					Use Premise			
											Perpendicular (Q, AB)				
					∠QOB △QOB≡ ?										
											Perpendicular (Q, DC)				
				△DLQ≡△QOB QO⊥AB&QL⊥DC											
													△DLQ≡△QOB		
				QO=LQ △DLQ≡△QOB											
													DM=MA		
													QO=LQ		
														DM=MA	
					△DLQ≡△QOB										

Table 3: Overall performance of CK solving problem #1. (Cont.)

Reference				Forward Chaining			Backward Chaining			Construction		Heuristics			
4.11	33.33	200.00	50.00	800.00	50.00	25.00	400.00	125.00	260.00	23.08	300.00	44.44	250.00	40.00	25.00
3	1	2	1	8	4	1	4	5	13	3	9	4	10	4	1
READ-given	REF-goal	FOCUS	SEEK-axiom	FC	SEEK-fc	ASSUMPTION-fc	BC	SEEK-pattern	WISH	SEEK-const	Construction	sowhat	helpless	NotActually	STRATEGY
							Parallel								
								7							
					AP=PC										
												AP=PC			
											Seq(PB)				
					Seg(PB)										
												Seg(PB)			
											Seq(PB)				
													Seg(PB)		
							8								
				∠MPA=∠RPQ											
													∠MPA=∠RPQ		
					9	∠MPA=∠DCA or ∠CDQ=∠DQM △DCR∽△RPQ ∽△MPA △DCR∽△RPQ									
				∠DCR=∠CAB (DC//AB)			10								
												∠DCR=∠CAB			
			11	∠DRC=∠PRQ (Vertex)											
										Construction					
								12	Construction Right Angle						
														Isosceles Tri	
				∠CDR=∠RBA (DC//AB)											
				△DCR∽△RAB	13										
													△DCR∽△RAB		
								Triangle wit MQ							
										Construction					
								14							
											Perpendicular (D, AB)				
		△DAB, △DAT													
													△DAB∽△DAT		
											QA				
												15			
									∠DQA=∠DAB						
													∠DQA=∠DAB		
												16			
									△DMQ∽△DAB						
													△DMQ∽△DAB		
								Bisector Segment							
								Midpoint Law							
											Extend(CQ) Extend(DP)				
									17						
													Extend(CQ) Extend(DP)		
									18						
								Midpoint Law							
											Extend(CQ) Extend(DP)				

Table 4: The lines of reasoning

	0									
1										
	1	Premises → Trapezoid								
	9	bisector	MQ//AB	Equal segments Equal angles	MQ//AB	Similar Triangles	Need Construction	Need Construction	Need Right Angle	Perpendicular (R,AB)
2										
	0									
	2			Equal Angles	(NotActually) ∠DMP=∠DAB					
3										
	0									
	1				(NotActually) ∠DMB=∠DMP or ∠DAB					
4										
	0									
	2			∠DQM=∠CDQ	Triangle with ∠DQM, ∠CDQ					
5										
	0									
	3				(sowhat) Use Premises	Perpendicular (Q, AB)		Perpendicular (Q,DC)		
6										
	3						∠QOB=∠R → △QOB≡△?		QO⊥AB, QL⊥DC → △DLQ≡△QOB (helpless)	△DLQ≡△QOB → QO=LQ (helpless)×3
	3		Parallel	(sowhat) use premises AP=PC	(helpless)×2 Seg(PB)					
7										
	0									
	0									
8										
	1	∠MPA=∠RPQ (helpless)								
	0									
9										
	1	If ∠MPA=∠DCA or ∠CDQ=∠DQM → △DCR∞△RPQ ∞△MPA								
	0									
10										
	1	DC//AB → ∠DCR=∠CAB (sowhat)								
	0									
11										
	1	Vertex → ∠DRC=∠PRQ								
	1	Right Angle								
12										
	0									
	0									
13										
	2	DC//AB → ∠CDR=∠RBA	∠DRC=∠PRQ → △DCR∞△RAB (helpless)							
	3	Triangle with MQ	Need construction	Perpendicular (D, AB)						
14										
	1				△DAB∞△DAT (helpless)					

Table 4: The lines of reasoning (Cont.)

			(helpless)							
	1		∠DQA=∠DAB							
15										
	1	QA								
			(doesn't help construction)							
	1		△DMQ∽△DAB							
16										
	0									
				(helpless)						
	3	Bisector Segment	Midpoint law	extend(CQ) extend(DP)						
17										
	0									
	2	Midpoint law	extend(CQ) extend(DP)							
18										
	0									

Table 5: Pattern of postulate applications.

	0									
1										
	1	trapezoid								
	9	midpoint	tri-midpoint	?	tri-midpoint	similar triangle	?	?	similar-triangle	?
2										
	0									
	2			?	F-theorem					
3										
	0									
	1				F-theorem					
4										
	0									
	2			Z-theorem	triangle- Congruent					
5										
	0									
	3				Use Theorem	?		?		
6										
	3						?		right-angle triangle congruent	triangle-congruent
	3		tri-midpoint	Use Premise	?					
7										
	0									
	0									
8										
	1	vertical angle								
	0									
9										
	1	similar triangle								
	0									
10										
	1	Z-theorem								
	0									
11										
	1	Vertical Angle								
	1	?								
12										
	0									
	0									
13										
	2	Z-threom	similar triangle							
	3	?	?	?						
14										
	1				similar triangle					
	1		?							
15										
	1	?								
	1		similar triangle							
16										
	0									
	3	Bisector Segment	Midpoint law	?						
17										
	0									
	2	Midpoint law	?							
18										
	0									