PCLEARN: A model for learning perceptual-chunks

Masaki Suwa **Hiroshi Motoda**
Advanced Research Laboratory, Hitachi Ltd.
2520, Hatoyama, Saitama, 350-03, Japan
{suwa,motoda}@harl.hitachi.co.jp

Abstract

Past research in cognitive science reveals that prototypical configurations of domain objects, called *perceptual-chunks*, underlie the abilities of experts to solve problems efficiently. Little research, however, has been carried out on the mechanism used for learning perceptual-chunks from solving problems. The present paper addresses this issue in the domain of geometry proof problem-solving. We have developed a computational model that chunks, from problem diagrams, configuration of the elements which are visually grouped together, based on *perceptual* chunking criterion. This criterion, called *recognition rules*, reflects how people see problem diagrams and thus works effectively to determine which portion of problem diagrams are more likely to be grouped as a chunk. This distinguishes the proposed method from the *goal-oriented* chunking techniques used in machine-learning community. Experiments on solving geometry problems show that our technique can detect essential diagram configurations common to many problems. Additionally, implications of the *recognition rules* are discussed from a cognitive point of view.

Introduction

Characterizing the adaptive processes that allow for the acquisition of expertise is a significant goal of cognitive science. In the past there have been extensive studies on expert-novice distinctions in many aspects. One aspect of this is, as De Groot (1966) showed, chess masters exhibit better recall of realistic board positions than do novices, but not randomly placed board positions. Similar findings were presented in other domains as well, such as in the sequence of a baseball game (Voss et al., 1980) and in an electric circuit diagram (Egan & Schwaltz, 1979). Another aspect is that experts show a greater tendency to reason forward from the given conditions of a problem rather than backward from the goal (Patel & Groen, 1986). In other words, they react to some features within a problem for devising appropriate plans, without resorting to goal-directed backward planning. Agre calls this *reasoning by reacting* (Agre & Chapman, 1987). Expert's reactive performance is observed in the task of x-ray film perception as well (Kundel & Nodine, 1983); their visual attention tends to be immediately directed toward the parts containing abnormalies, without searching through the entire film. These aspects of expertise are attribited primarily to the cognitive structures for storing prototypical configurations of objects about which reasoning is performed. They are called *perceptual-chunks* (or *schemas*).

How, then, do people learn perceptual-chunks through experience that allow them to transform themselves from novices to experts? This is called a schema acquisition problem. Surprisingly, little research has been carried out on this issue. Sweller (1988) attributed this lack of activity to the conventional framework of goal-directed problem-solving, such as means-ends analysis. He pointed out that the restricted focus on reducing differences between a given problem state and the goal state under a means-ends strategy deprives problem-solvers of both the intention and cognitive processing capacity necessary to detect essential schematic features in problem structures. Most learning systems have been necessarily *goal-oriented* as well, since they learn from problem-solving traces that goal-directed solvers make. For example, SOAR (Laird et al., 1987) chunks in a goal-oriented way based on *universal subgoaling* technique, and various explanation-based learners (Mitchell et al., 1986), in concept formation and macro-operator learning, chunk goal/subgoal structures that explain the solver's target concepts. Although goal-orientedness is preferably general and domain-independent, it follows at the same time that those learners fail to reflect human regularity in feeling which portions of problem structures are more likely to be chunked into schemas. The mechanism of chunking them seems more perceptual and more specific to the domain objects themselves appearing in problem-solving situations (Koedinger & Anderson, 1989).

This paper presents a computational model called the PCLEARN (perceptual chunking learner) that learns perceptual-chunks in geometry proof problem-solving, based on a *perceptual* criterion that reflects human regularity in perceiving and detecting essential features in problem strcutures. Geometry proofs are suitable for this study because solving and learning from geometry problems involves recognizing diagrammatic information of problems perceptually. In the second section, we discuss regularity in experts' solving and learning geometry problems. The third section explains our perceptual-chunking method and investigates its feasibility from results provided by our model. In the fourth section, we argue that learning perceptual-chunks is relevant to constructing "good" and "meaningful" mental models that may facilitate efficient and flexible problem-solving performance.

How people solve and learn in geometry domains

As discussed in many studies on geometry proof problem-solving (Greeno, 1983; McDougal & Hammond, 1992; Koedinger & Anderson, 1990), perceptual chunks underlie the ability of experts to solve and learn efficiently; experts solve problems by matching familiar chunks to the subparts of diagrams representing entire problems (Koedinger & Anderson, 1990), and they then extract certain essential subdiagrams as new perceptual-chunks. This latter process of decomposing the entire diagram into subdiagrams for perceptual-chunks either used or newly extracted is essential for understanding problem structures.

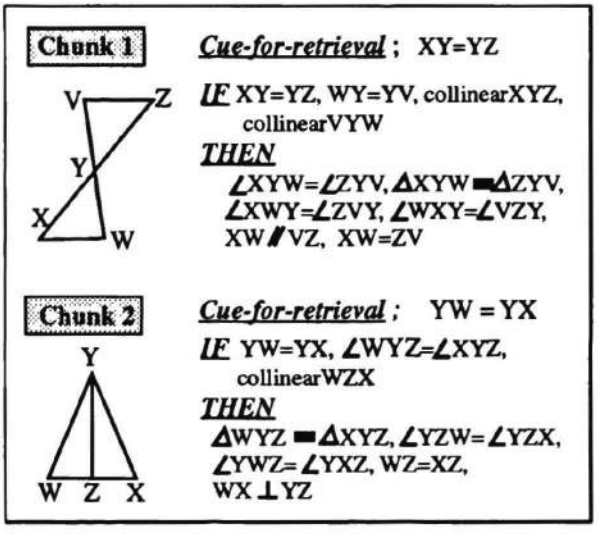

Figure 1: Typical perceptual-chunks in the domain of geometry

Figure 1 shows diagrams for two typical perceptual-chunks with their corresponding macro-operators. These macro-operators are needed for these chunks to be applied to problem-solving situations; each chunk is recalled by the existence of a certain problem feature that matches cue-for-retrieval information and, thus, the corresponding macro-operator is tested for application. If all the statements in the IF part are satisfied within the current problem, the instantiated statements of the THEN part will create new nodes in the proof tree.

What portions of the problem diagrams are more likely to be chunked as essential information into perceptual-chunks? In other words, what kind of regularity do experts reveal in detecting and chunking problem features? Koedinger et al. (1990) presented evidence from verbal reports on subjects indicating that, when subjects applied perceptual-chunks, they could not always immediately recall what intermediate macro-operator sequences the chunks were configured from. It is more likely that experts *perceptually* chunk an aggregation of domain objects appearing in problem-solving situations, not sequences of consecutively applied problem-solving operators, and that they remember the diagrams corresponding to the chunked objects.

This regularity theory is consistent with observations that even experts cannot solve problems without drawing diagrams of the problem involved. Why diagrams are essential is a common question among all studies on diagrammatic reasoning (Narayanan et al., 1992). One possibility may be that diagrammatic information cues relevant perceptual-chunks in a current problem-solving context because they are memorized as configurations of diagrammatic features. If so, regularity in acquiring perceptual-chunks may arise from *seeing* diagrammatic features in a current geometry problem.

Perceptual-chunking method

We designed our model PCLEARN to chunk the diagram elements of a problem that are *visually grouped together*, when it has solved the problem. We hypothesized that domain-specific knowledge about how people *see* diagrammatic features of domain objects determines which elements should be visually grouped together, and called the knowledge *recognition rules*. Extraction of diagram configuration by use of *recognition rules* is performed in process 2 described later. In order for acquired chunks to be really useful in future problem-solving situations, PCLEARN refers to the proof-tree of a target problem in the following objectives; (1) to detect seeds that motivate chunking processes (process 1), (2) to provide a macro-operator to the extracted diagram elements (process 3), (3) and to remove irrelevant portions from the diagram elements of an extracted chunk so that it will be meaningful in problem-solving contexts (process 3).

The main point that distinguishes the PCLEARN system from conventional learning systems, including EBL systems, is the use of *recognition rules* as a chunking criterion. Conventional systems chunk such problem-solving traces that lead to a certain target concept (Mitchell et al., 1986; Minton et al., 1989) or have been developed under a subgoal in the goal-structure hierarchy (Laird et al., 1987). Consequently, if we have a careful look at the learned chunks, they are not always identical with those to be acquired perceptually by human experts.

Recognition rules

A domain object, O, is "recognizable" when people become aware of its existence in chunking processes, and this statement is represented as a literal *recognizable*(O). Each recognition rule describes the conditions necessary for a domain object to be recognizable. Its head is a literal expressing that a target object is recognizable, and its bodies are conjunctions of (1) the literals expressing that other *related* objects are recognizable and (2) the literals expressing that *additional conditions* hold among those objects. By *related*, we mean part-whole relationships. For example, the domain objects we deal with in geometry are points, segments, angles and triangles. Part-whole relationships are such that a point is a part of a segment, a segment is a part of an angle or a triangle, and so on. In the first case we can say that the point and the segment have

recognizable(X):- recognizable(s(X,Y)).
recognizable(s(X,Y)):- recognizable(a(X,Y,Z)).
recognizable(s(X,Y)):- recognizable(tr(X,Y,Z)).
recognizable(s(X,Y)):-
recognizable(X), recognizable(Y), exist(s(X,Y)).
recognizable(s(X,Y)):-
recognizable(X), recognizable(Y), collinear(X,Z,Y).
recognizable(a(X,Y,Z)):-
recognizable(s(X,Y)), recognizable(s(Y,Z)).
recognizable(tr(X,Y,Z)):-
recognizable(s(X,Y)), recognizable(s(Y,Z)),
recognizable(s(Z,X)).

where *s(X,Y)* -- segment XY, *tr(X,Y,Z)* -- triangle XYZ,
a(X,Y,Z) -- angle XYZ
The literals underlined are *additional conditions.*

Figure 2: The set of recognition rules used in geometry

a part-whole relationship, and in general, whole objects in a domain constitute a partially ordered hierarchy in terms of part-whole relationships.

Figure 2 is the set of recognition rules we provided for geometry. The first rule states that point X is always recognizable when segment XY is recognizable. When an object is recognizable, its partial objects will be always recognizable as well. The first three rules listed in Fig. 2 pertain to this category. In contrast, when certain partial objects of an object are recognizable, we cannot always determine recognizablity of the object and sometimes need some *additional conditions*. For example, for segment XY to be recognizable, it does not suffice to confirm the recognizability of the two end points X and Y; additionally we have to prove the statement that segment XY actually exists in the problem diagram ($exist(s(X,Y))$) or the statement that segments XZ and ZY are on the same line for another point Z ($collinear(X,Z,Y)$).

Each rule only expresses the interrelationship between the recognizability of related objects. Therefore, recognition rules can be used to determine the entire set of recognizable objects only when statements for certain recognizable objects are initially given (see Process 2 for the details of their use).

Perceptual-chunking processes

Perceptual-chunking is performed after the solver finishes solving a problem, i.e. constructing a full proof-tree. It consists of the following four processes.

Process 1: Detecting seeds for chunking

The first process involves identifying, in the proof-tree, seeds which motivate chunking processes. The PCLEARN solver searches for applicable problem-solving operators and/or perceptual-chunks it has already retained in order to produce an AND/OR proof tree[1]. A seed node for learning is the one which is relevant to proving the goal node and also to which irrelevant problem-solving operators have been tested and/or applied[2]. This node is called a *control decision node* (CDN), and the relevant problem-solving operator applied at a CDN is called SAO (successfully applied operator). The notion of CDN itself is not original; this corresponds to a training example selected by the "other-operators-have-failed" heuristic in PRODIGY systems (Minton et al., 1989).

CDNs are promising for learning seeds because, if a problem-solving search at a CDN is directed well by using a relevant perceptual-chunk, the solver may save the cost of applying other irrelevant operators. The subsequent three processes will be performed for each CDN identified. Figure 3 is an example of the proof tree for problem 1 in the set of problems shown in Appendix A. Only relevant nodes and links are represented here, *not* those nodes produced but irrelevant to the goal *nor* those operators that have been tested but not applied. The underlined nodes here are the CDNs. We explain the subsequent process for the CDN, $AC = CF$.

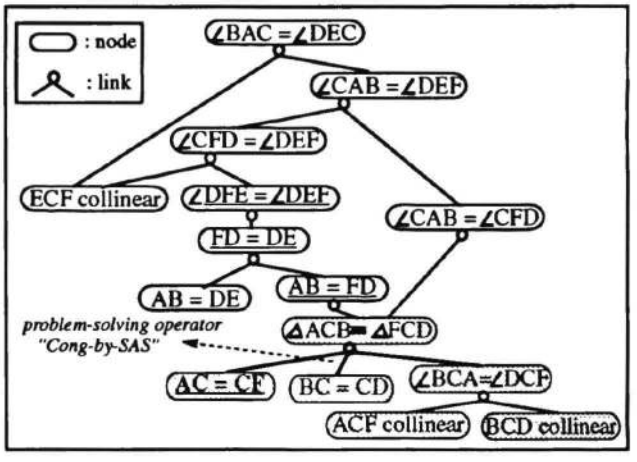

Figure 3: A proof-tree for Problem 1

Process 2: Extracting diagram configuration as a chunk

Chunking for a CDN starts by *seeing* the diagram elements included in the SAO for the CDN. Technically this means asserting that the domain objects appearing as arguments of the constituent statements for the SAO are all recognizable as initial givens. Process 2 involves extracting the diagram configuration composed of those domain objects which are *grouped in a chunk together* with the above recognizable objects in the SAO. We begin by matching the recognizability of SAO-involving objects to the bodies of recognition rules and finally produce the entire set of recognizable domain objects.

[1]The tree consists of nodes and links; the nodes represent the statements given initially or produced during the proof, and a link between several nodes and their upper node represents an operator used for proving the existence of the upper node when the lower nodes exist. The node on top of the tree represents the goal statement.

[2]During problem-solving search, in general, many trials of testing to apply the available problem-solving operators occur, most of which will fail in vain.

The SAO for $AC = CF$ is the theorem of congruence by SAS (two sides and one angle) (see Fig. 3). Objects BC, AC, CD, CF, $\angle BCA$, $\angle DCF$, $\triangle ACB$ and $\triangle FCD$ are involved in the SAO and these are initially asserted as recognizable. Use of recognition rules, then, proves the recognizability of the following objects; A, B, C, D, F, AB, DF, BD, AF, $\angle BAC$, $\angle BAF$, $\angle ABC$, $\angle ABD$, $\angle DFC$, $\angle DFA$, $\angle FDC$, $\angle FDB$, $\angle BCF$ and $\angle ACD$. The entire set of extracted recognizable objects forms a diagram configuration.

Process 3: Providing a macro-operator to the diagram configuration

This process begins by identifying the following statements in the proof tree; (1) the statements of the *additional conditions* appearing in the recognition rules used in process 2, only when they were verified, and (2) the statements constituting the SAO. These two kinds of statements are the ones that human beings would recognize when they *see* the diagrams elements of the SAO. Especially the first ones are important because they are implicitly included in the current appearance of the extracted diagram configuration and thus they will be essential features constituting the macro-operator to be provided. Examples of the first are *collinearACF* and *collinearBCD* verified in proving the recognizability of AF and BD. Examples of the second are $AC = CF$, $BC = CD$, $\angle ACB = \angle FCD$ and $\triangle ACB \equiv \triangle FCD$. These are shown in Fig. 3 as grey nodes.

The statements located lowest on the proof-tree out of all these identified ones will form the antecedent part of the macro-operator to be attached to the extracted diagram configuration. In our example, $AC = CF$, $BC = CD$, *collinearACF* and *collinearBCD* correspond to these. The intermediate and the remaning parts of the macro-operator are obtained by deriving all the possible statements in the restricted environment where only the above identified statements hold within the diagram extracted in process 2. By this derivation, we obtain $AB = FD$, $\angle CAB = \angle CFD$, $\angle CBA = \angle CDF$ and $AB//DF$. The last two should be incorporated in a chunk although they were not relevant in the current proof. This derivation process is essential also for the following objective. Those diagram elements that do not appear during this process are judged to be irrelevant to problem-solving contexts and thus removed from the diagram configuration. The new diagram configuration developed here is the final configuration of the perceptual-chunk to be acquired.

No diagram elements were removed in the examples we have shown. A good example of this is the chunking process for the CDN, $AB = FD$. In this case, the extracted diagram configuration in process 2 includes segment AF with the collinear statement *collinearAEF*. Because the collinearity is irrelevant in obtaining the macro-operator, however, segment AF will be removed and thus the constraint forcing point A on the same line as segment EF will be eliminated.

Process 4: Generalizing

The final process involves generalizing each node in the macro-operator sequences by dissolving problem-specific instantiations of each constituent operator. The generalized version of the macro-operator will be attached to the diagram configuration acquired in process 3. The left perceptual-chunk in Fig. 1 is acquired for the CDN, $AC = CF$.

Note here that use of recognition rules restricts the derivation in process 3 to a portion of the entire proof-tree and thus determines which portion of the entire problem diagrams should be chunked into a perceptual-chunk.

Experimental Results

A desirable computational model that learns perceptual-chunks should provide for the identification of visual cues common to different problems. The following experiment supports the feasibility of our model. We selected twenty geometry problems belonging to two categories of geometry proofs[3], expecting that they would reveal cross-problem commonalities in terms of chunks. The whole set of problems used for this experiment is shown in Appendix A. PCLEARN solves each problem in ascending order of problem complexity[4]. After solving each one, it acquires perceptual-chunks and stores them. In solving problems, the solver is allowed to use the perceptual-chunks acquired from solving the previous ones[5]. After solving all twenty problems, we examined the frequencies at which the same perceptual-chunks in terms of diagrammatic information[6] are learned from solving different problems.

Table 1: The frequencies of perceptual-chunks being learned

Frequency	The number of perceptual-chunks	
	PCLEARN	EBL
1	43	86
2	9	7
3	5	0
4	3	1
more than 4	4	0
total	64	94

We compared the result for PCLEARN with that for an explanation-based learner (Minton et al., 1989) which chunks the inference path that reaches out from each CDN to the goal node. Table 1 shows the number of the

[3] *Parallel lines and angles* and *congruence of triangles*

[4] For simplicity, we use the sum of the numbers of lines and points as a measure of complexity.

[5] Strictly speaking, the preferential order of macro-operators being selected for use will affect problem-solving performance as well as the kind of chunks to be acquired in future. Analysis of *cost-effective utility* of learned knowledge may be the best to provide ordering to the entire set of macro-operators. Readers can refer to (Suwa & Motoda, 1993) for further information on this.

[6] Various macro-operator sequences are possible in a diagram configuration.

kinds of perceptual-chunks for each frequency at which they were acquired. The total number of chunks acquired from different problems more than two times are 21 in PCLEARN and 8 in EBL. This suggests that the *perceptual* chunking technique provides advantages over *goal-oriented* chunking in extracting essential diagrammatic configurations common to different problems.

As for improvement of problem-solving performance by use of perceptual-chunks, we have papers (Suwa & Motoda, 1993; 1994) on cumulative costs for solving the twenty problems in PCLEARN, the EBL system and the system without any learning module. A brief characterization of the comparison is that we obtain better problem-solving performance in PCLEARN than the system without learning, but heavy degradation in the EBL systems. This is mainly because the chunks learned by EBL tend to be specific to the goal-structure of the original problem and thus have extremely low applicability to other problems. In contrast to that, the chunks learned by PCLEARN are such ones that can be recognized commonly in many problems and thus have higher applicability. See details in the above references.

General Discussions

If we assume that perceptual-chunks are domain-specific visual categories acquired through experiences, arguments may arise on whether they should be acquired as "visually good chunks" determined by gestalt principles (Bower & Glass, 1976), or as chunks "meaningful" to theoretical knowledge and problem-solving planning. This is an important but open question. For X-ray film recognition (Kundel & Nodine, 1983), theoretical knowledge about organs, cancer and so on helps us decompose the entire film area into chunks. In chess domain (Chase & Simon, 1973), chunks are configured so that they reflect plans in attack and defence. This means that they are meaningful chunks. In contrast, typical gestalt laws that control the process of perceiving line drawings are *common direction* and *proximity* (Bower & Glass, 1976). Gestalt laws say that two diagram elements satisfying as a whole either feature are likely to be chunked together.

In our model, as suggested in section 3, the recognition rule containing a *collinear* statement plays an important role in perceptual-chunking. Using this rule and incorporating collinearlity as a feature in a perceptual-chunk has exactly the same connotation as recognizing two segments with a *common direction* as a chunk. In this sense, the PCLEARN model has an aspect of chunking "visually good" subdiagrams. It also has an aspect of chunking "meaningful" diagrams in the three ways mentioned earlier; it uses information on problem-solving traces for detecting motivation from which to learn, for assigning a macro-operator to the extracted diagram configuration, and for removing the diagram elements that are irrelevant to problem-solving knowledge. As for the effect of eliminating irrelevant diagram elements, it would be valuable to do a sensitivity analysis of the processes in PCLEARN for evaluating which processes are really central for acquiring "meaningful" perceptual-chunks. That is one of our future work.

Related to this issue are the past extensive psychological experiments in which subjects are exposed to a whole diagram and are supposed to recall subdiagrams or recall the diagram from some partial cues (Bower & Glass, 1976; Reed, 1974). Here, configurations natural to human eyes provide good visual cues. These experiments, however, were done in the context of looking at diagrams without objectives or intentions. Rather how subjects see diagrams in the context of problem-solving objectives may be an interesting issue for psychological investigation. That is one of our future work in this domain because geometry proofs are suitable for this investigation.

Recognition rules can be extended so as to apply in many domains as a perceptual-chunking criterion. Recognition rules are the knowledge representing how human beings see diagrammatic features of domain objects. Since people with different levels of knowledge differently recognize problem structures, PCLEARN could potentially model different skill levels by manipulating the sophistication of the rules.

Visually decomposing the problem diagrams into subparts is relevant to constructing mental models (Johnson-Laird, 1983) and recognizing analogy across problems. In order to establish a theory that allows for the construction of mental models which are good, natural and meaningful and for efficient analogical reasoning, further work along these lines must be pursued.

Conclusion

We developed a computational model PCLEARN that learns prototypical configurations of diagram elements, called *perceptual-chunks*, in the domain of geometry proof problem-solving. Our model *perceptually* chunks, for each of the seed nodes in the proof-tree, the portion of problem diagrams that are visually grouped together with the diagrams corresponding to the problem-solving operator applied to the seed node. We use a perceptual chunking criterion, called *recognition rules*. This criterion is domain-specific knowledge about how people *see* problem diagrams and thus works effectively to determine which portion of diagrams should be visually grouped together. This distinguishes our chunking method from other theories in the machine learning community which chunk knowledge in terms of goal/subgoal structures in problem-solving traces. The acquired chunks in our model are meant to be "visually good and natural" because PCLEARN uses perceptual criterion, and are also meant to be also "meaningful" in problem-solving situations, because it uses information from an explored problem-solving proof tree for providing relevant macro-operator and for removing irrelevant diagram elements from the extracted configuration.

Acknowledgements

We thank anonymous reviewers for giving us many instructive comments.

References

Agre, P. & Chapman, D. (1987). Pengi: An implementation of a theory of activity. In *Proc. of AAAI-87* (pp. 268–272).

Bower, G.H. & Glass, A.L. (1976). Structural units and the redintegrative power of picture fragments. *J. of experimental psychology: Human learning and memory*, 2(4), 456–466.

Chase, W.G. & Simon, H.A. (1973). Perception in chess. *Cognitive Psychology*, 4, 55–81.

De Groot, A. (1966). Perception and memory versus thought: some old ideas and recent findings. In Kleinmuntz, B. (Eds.), *Problem solving*. Wiley.

Egan, D. & Schwaltz, B. (1979). Chunking in recall of symbolic drawings. *Memory and Cognition*, 7, 149–158.

Greeno, J. G. (1983). Forms of understanding in mathematical problem-solving. In *Learning and Motivation in the Classroom*. Erlbaum.

Johnson-Laird, P.N. (1983). *Mental Models*. Harvard University Press.

Reed, S.K. (1974). Structural descriptions and the limitations of visual images. *Memory and Cognition*, 2, 329–336

Koedinger, K.R. & Anderson, J.R. (1989). Perceptual chunks in geometry problem-solving: a challenge to theories of skill acquisition. In *Proc. of the 11th Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum Associates.

Koedinger, K.R. & Anderson, J.R. (1990). Abstract planning and perceptual chunks: elements of expertise in geometry. *Cognitive Science*, 14, 511-550.

Kundel, H.L. & Nodine, C.F. (1983). A visual concept shapes image perception. *Radiology*, 146, 363–368.

Laird, J.E., Newell, A. & Rosenbloom, P.S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence*, 33(1), 1–64.

McDougal, T. & Hammond, K. (1992). A recognition model of geometry theorem-proving. In *Proc. of the 14th Annual Conference of the Cognitive Science Society* (pp. 106–111).

Minton, S., Carbonell, J.G., Knoblock, C.A., Kuokka, D.R., Etzioni, O. & Gil, Y. (1989). Explanation-based learning: a problem solving perspective. *Artificial Intelligence*, 40, 63–118.

Mitchell, T.M., Keller, R.M. & Kedar-Cabelli, S.T. (1986). Explanation-based generalization : a unifying view. *Machine Learning*, 1(1), 47–80.

Narayanan, N.H. et al. (Eds.). (1992). *Working Notes of the AAAI Spring Symposium on Reasoning with Diagrammatic Representations.*

Patel, V.L. & Groen, G.J. (1986). Knowledge-based solution strategies in medical reasoning. *Cognitive Science*, 10, 91–116.

Suwa, M. & Motoda, H. (1993). A perceptual criterion for visually controlling learning. In *Proc. of 4th international workshop on algorithmic learning theory*, (pp. 356–369). Lecture Notes in AI 744, Springer-Verlag.

Suwa, M. & Motoda, H. (1994). PCLEARN: A computer model for acquiring perceptual-chunks. to appear in *AI Communications*, The European journal on artificial intelligence, June issue.

Sweller, J. (1988). Cognitive load during problem-solving: effects on learning. *Cognitive Science*, 12, 257–285.

Voss, J.F., Vesonder, G.T. & Spilich, G.J. (1980). Text generation and recall by high-knowledge and low-knowledge individuals. *J. of verbal learning and verbal behavior*, 19, 651–667.

A. The geometry problems used for experiments