Memory for Problem Solving Steps

Stellan Ohlsson

Learning Research and Development Center
University of Pittsburgh,
Pittsburgh, PA 15260, USA
stellan@vms.cis.pitt.edu

Abstract

A widely adopted theory of procedural learning claims that people construct new problem solving rules through induction over past problem solving steps. The underlying assumption that people store information about problem solving steps in memory was tested by measuring subjects' memory of their own problem solving steps in four different ways. The results support the assumption that people store enough information in memory to enable induction of new problem solving rules.

Memory in Procedural Learning

Theoretical analyses of procedural learning have converged on three general principles for the acquisition of problem solving rules through practice: (a) Learners solve unfamiliar problems through weak problem solving methods such as hill climbing, planning, and search. The function of a weak method is to generate task relevant behavior in the absence of knowledge about the task. (b) Information about the problem solving steps generated by the weak method--context, type of action, outcome--is stored in long-term memory. (c) New problem solving rules are induced from the resulting memory record.

This general theory of procedural learning has been instantiated in several computer simulation models. The models differ with respect to problem solving method, type of information stored in memory, and induction mechanism. For example, the ACT* model (Anderson 1983) solves problems through planning, stores information about the temporal order of problem solving steps, and learns through rule composition (as well as in other ways). The UPL model (Ohlsson 1987) solves problems through search, stores state-operator pairs in memory, and learns through generalization (as well as in other ways). The SAGE model (Langley 1987) solves problems through search, stores information about good and bad outcomes, and learns through discrimination. The Soar model (Newell 1990) also solves problems through search, stores sequences of search steps, and learns through chunking. Although these (and many other) models of procedural learning differ in their specifics, they nevertheless instantiate the same three general principles stated above.

The most common approach to empirical validation of such models is to test their *predictions* against data. For example, validation might involve deriving the power law of learning (Newell & Rosenbloom 1981) or observed error patterns (VanLehn 1990) from a given model. An alternative approach is to provide empirical support for the *assumptions* underlying the entire class of models, preferably support which is independent of the learning data which inspired the creation of the models.

One central assumption of the theory of procedural learning is that people store detailed information about their own problem solving steps in memory while solving unfamiliar problems. Because procedural learning is most rapid at the outset of learning, the hypothesis of internal induction implicitly assumes that the memory record builds up rapidly, i. e., that the memory record is not only detailed but also relatively complete. This assumption is dubious. Both problem solving and memory storage are capacity demanding processes, so it is reasonable to expect them to interfere with each other. If it can be shown that learners do not store information about problem solving steps in memory, the plausibility of the the current theory of procedural learning, and of the entire class of learning models that build on it, is undermined. Conversely, if it can be shown that people indeed store a detailed and complete memory record of their own problem solving steps, the plausibility of the theory of procedural learning is strengthened.

In a study of the Missionaries and Cannibals Puzzle, Reed and Johnsen (1977) found relatively good recognition memory for problem states (63% correct recognition, on the average), but rather poor ability to recall which step was taken in a given problem state. However, as Reed and Johnsen (1977, p. 198) point out, measuring memory for problem solving steps is not straightforward. Unlike the stimuli used in ordinary memory experiments, problem solving steps are generated by the subject, rather than by the experimenter. Hence, the experimenter cannot prepare the stimuli in advance. Problem solving steps can be recorded and used as stimuli later. But accurate assessment of the memory trace has to occur immediately after problem solving, before the trace is obliterated by decay or interference.

In the present study, the power of the computer was used to overcome this methodological difficulty. The experimental problems were presented to the subject via a computer terminal. The computer recorded the subjects' steps. Immediately after a subject indicated that he[1] was finished with a problem, the computer retrieved the subject's solution path, constructed a memory test on the basis of it, and presented the test to the subject.

General Method

The subjects' memory for their own problem solving steps were measured in different ways in four experiments which followed the same general format.

Problem Solving Tasks

In the type of problem used in this study, the subject was shown three unmarked jars of different capacities. Initially, the largest jar is full of water and the two smaller jars are empty. The task is to redistribute the water so that the two larger jars have equal amounts (and the smallest jar is empty). The only available action is to pour water from one jar into another jar, until either the source jar is empty or the receiving jar is full, whichever happens first. A solution consists of a sequence of pouring actions that transforms the initial distribution of water into the desired one. Problems like these are known as *water jar problems.* They were introduced into the psychological laboratory by Luchins (1942) and have been studied frequently (Atkinson, Masson, & Polson 1980).

The subjects in all four experiments solved the same six water jar problems in the same order. The simplest problem (with jar capacities 6, 5, and 3) required only two steps and served as a training problem. The second problem was the 8-5-3 problem studied by Atkinson, Masson, and Polson (1980). The remaining four problems had capacities 16-10-6, 24-15-9, 32-20-12, and 40-25-15, respectively. They are isomorphs of the 8-5-3 problem, i. e., all five experimental problems had exactly the same problem space. The graph of this space has 16 states and the shortest path to solution is eight steps long (see Atkinson, Masson, & Polson 1980, p. 183). There are many opportunities to unintentionally return to previously visited states in this space, making the problems non-trivial for novices.

Presentation System and Procedure

The problems were presented on a computer terminal. The computer represented a problem in an alphanumeric, as opposed to graphical, format: Each jar was labelled with a single letter, and the computer showed a problem state by printing a table with the letters and the current water volumes of the corresponding jars on the screen. The first screen showed both the initial state and the goal state; subsequent screens did not show the goal state but only the current state. Although the initial state could be reviewed at any time (see below), there was no way to review other past problem states without returning to them.

The subject indicated which pouring action he choose by first pressing the keyboard key corresponding to the jar he was pouring from and then the key corresponding to the jar he was pouring into. Illegal moves, e. g., attempting to pour from an empty jar, were rejected with a short explanation (e. g., "You cannot pour from an empty jar").

When the subject reached the goal state, the computer system waited for the subject to press a key that indicated that he knew that he was done with the problem. If the subject instead performed yet another pouring action, thus moving away from the goal state, the system made no objection. The subject could also press a key that indicated that he gave up. The subject also had the options of backing up, i. e., returning to a state preceeding the current state, and of restarting, i. e., of returning to the initial problem state. The initial problem state could also be reviewed at any point along the way. Finishing, giving up, backing up, restarting, and reviewing were each signalled by a single key press, except backing up, which required a second key press indicating the number of steps the subject wished to back up.

When the subject signalled that he was done with a problem, the system constructed a memory test based on its recording of the subject's behavior. The memory items were then presented one at a time. On each item, the subject was asked for a response--different for different memory tests--and a confidence judgment. The confidence judgment was made on a five point scale, from 1 ("just guessing") to 5 ("completely certain"). The subject was then given feedback about the correctness of his response, and the next memory item was presented. After the memory test was completed, the next problem solving task was presented, and so on, through the five trials. Each trial thus consisted of one problem solving attempt followed by a memory test.

The subject was given a brief description of the purpose of the experiment, and he was told that he would be subject to a memory test after each problem. The instructions emphasized that the memory test should be answered by recalling what was done during problem solving, rather than by re-thinking the problem solving task. The experimenter explained the idea behind water jar problems and the operation of the computer system, and remained in the laboratory while the subject solved the training problem and the memory test that followed it. He then waited outside the laboratory while the subject

---

[1] For the sake of brevity, the pronouns "he", "his", "himself, etc. are used generically to mean "he or she", "his or her", "himself or herself", etc.

solved the five experimental problems and responded to the associated memory tests.

Subjects

Eight subjects, four psychology students and four computer science students, participated in each of the four experiments, for a total of 32 subjects. The psychology students participated in order to fulfill a course requirement, while the computer science students were payed a small amount for their participation. Half the subjects were men and half were women. The two different subject populations were included in the experiment to ensure a wide range of puzzle solving ability among the subjects.

Evidence for Procedural Learning

The purpose of the study was to measure memory for problem solving steps *during procedural learning*. Hence, it is important to verify that that the subjects' problem solving performance improved across the five trials. Percent correct, time to solution, and number of steps to solution are commonly used performance measures. Percent correct does not accurately reflect knowledge about water jar puzzles, because the goal state can be found by exhaustive search, or even by a random walk, through the (rather small) problem space. Time to solution is only applicable on correctly solved problems. Worse, because different (correct) solutions have different lengths, time to solution confounds the question of which solution the problem solver knows with the question how fast he is executing it. Neither measure is an accurate indicator of procedural learning. Number of steps to solution have similar disadvantages.

The acquisition of a procedure for a puzzle problem should lead to better solutions. The *solution quality* was measured in the following way. If the subject gave up without having reached the goal state, his solution was assigned the value 0. If the subject reached the goal state, but not through the shortest path, then his solution was assigned the value 1. Shortest path solutions were assigned the value 2. Measured in this way, solution quality improved significantly across trials (Kruskal-Wallis one-way analysis of variance by ranks, corrected for ties; chi-square = 10.65, df = 4, $p < .05$).

A second consequence of procedural learning is that the problem solver has to think less in order to decide what to do in each problem state. Therefore, the *residence time*, i. e., the time the problem solver remains in a problem state before taking an action[2], ought to decrease. The computer system recorded the residence time for each subject and problem state. There was a statistically significant decrease in the average residence time across the five trials (two-way ANOVA, subject population x trials, F = 4.53, df = 4, $p < .01$).

In summary, both the quality of the solutions and the speed of decision making improved across trials, indicating that the subjects learned something about how to solve water jar problems in the course of the study.

Experiment 1: Path Retracing

The purpose of the first experiment was to measure the subjects' ability to retrace their own solutions. After the subject declared himself finished with a problem, the computer system retrieved the subject's solution path, presented the initial state of the problem and asked "Which action did you take in this situation?". If the subject responded with the action which he did, in fact, take while solving the problem, the system responded "Yes, that is what you did" and presented the next state on the subject's path. If the subject responded with an action different than the one he executed while solving the problem, the computer system answered "No, that is not what you did. Instead, you poured ... into ...", and presented the next problem state *on the subject's original solution path* (rather than the state that would result from applying the operator the subject responded with). In short, the *path retracing* test measured the subject's ability to recreate his solution by stepping through his path and asking him to recall which action he took in each successive problem state. Memory for state-operator pairs is crucial for learning mechanisms such as generalization (Anderson 1983; Ohlsson 1987) and discrimination (Langley 1987).

The results of this memory test are shown in Table 1, row 1. On the average, the subjects retraced 81% of their steps correctly. Although every subject performed better on trial 2 than on trial 1, Table 1 nevertheless suggests that the subjects' memory performance did not improve systematically across trials. A two-way ANOVA (subject population x trials) confirms that the effect of trials in Table 1 is not significant (F = 2.145, $p > .10$).

Experiment 2: Operator Recall

It is possible to solve the path retracing task used in Experiment 1 by re-solving the problem, rather than by retrieving the previous solution from memory. In order to prevent subjects from answering the memory items through problem solving rather than through memory retrieval, the subjects in Experiment 2 had to recall the operators in random order. After the subject indicated that he was finished with a particular problem, the computer system retrieved the subject's solution path, presented a *randomly* chosen state on

---

[2] The concept of residence time was introduced by Newell and Simon (1972, pp. 811-814).

Table 1. Memory performance, in terms of % correct, on each trial for each experiment (memory measure), averaged across subjects.

Memory measure	Trial 1	2	3	4	5	M
Path retracing	.62	.81	.86	.93	.82	.81
Operator recall	.72	.77	.74	.68	.90	.76
State recognition	.73	.84	.88	.83	.87	.83
Order recognition	.87	.86	.86	.87	.91	.87

the path, and asked "Which action did you take in this situation?". The system then presented another state selected at random, and so on, until the subject had responded to each state on his solution path. In short, the *operator recall* test measured the subjects' ability to recall which operator he executed in a randomly chosen problem state. This memory test is similar to the memory task used in Experiment II of Reed and Johnsen (1977).

The results of this memory test are also shown in Table 1, row 2. On the average, the subjects recalled 76% of their operators correctly. The memory performance did not improve systematically across trials (F =1.251, p > .30).

Experiment 3: State Recognition

In the third experiment the subjects were tested for memory of problem states rather than operators. After the subject indicated that he had finished with a problem, the computer system first retrieved all problem states on the subject's solution path. It then retrieved from the problem space an equal number of problem states which the subject had *not* visited while solving the problem. The two sets of states were mixed and arranged in random order. Thus, the subject was presented with a sequence of problem states, half of which he had seen while solving the problem and half of which he had not seen before (but which belonged to the same problem space). The subject responded by answering "yes" or "no" to the question whether he had encountered the shown state during problem solving. In short, the state recognition test measured the subjects' ability to discriminate the problem states they had visited during problem solving from those they had not. This memory test is similar to the recognition task used in Experiments I and II of Reed and Johnson (1977). Memory for problem states is crucial for learning mechanisms such as constraint violation (Ohlsson & Rees 1991) and subgoal discovery (Ohlsson 1987).

The results from this memory test are shown in Table 1, row 3. The subjects recognized 83% of the states correctly, on the average. There was no systematic improvement in state recognition across trials (F = 1.368, p > .10).

Experiment 4: Order Recognition

In the fourth experiment, the subjects were tested for memory of the temporal order in which they encountered particular problem states. After the subject indicated that he was finished with a problem, the computer system retrieved all the problem states the subject had visited during problem solving and constructed a set of memory items by pairing the problem states. The pairs were presented in random order. One state was presented on the left side of the terminal screen and the other on the right side. The computer system asked the subject "Which of these states did you encounter before the other?". In half the items, the left state was, in fact, encountered earlier in the solution path than the right state; in the other half, the reverse was the case. In short, the *order recognition* test measured the subjects' ability to remember in which order they encountered problem states. Memory for temporal order of problem solving events is crucial for learning mechanisms such as composition (Anderson 1983) and chunking (Newell 1990).

The results of this memory test are shown in Table 1, row 4. The subjects remembered the temporal ordering of 87% of the problem states correctly. As in the previous experiments, there was no significant training effect for the memory performance (F = .153, p > .10).

Summary and Discussion

The results of the four memory tests were uniformaly high, indicating that the subjects stored approximately 80% of the information probed for while solving the experimental problems. The performance was equally high across the four different ways of accessing memory: retracing a solution, recalling which operator was taken in a given problem state, distinguishing problem states encountered during problem solving from other states in the same problem space, and judging the temporal order in which problem states were encountered. Because the problem states and the operators in the water jar problem are visually impoverished and syntactically similar to each other, the interference effects ought to be strong, biasing the study against high memory performance. Hence, the results presented here might underestimate the amount of memory storage that is typical during problem solving.

The performance of the subjects in the present study is significantly higher than the performance of the subjects in Reed and Johnsen (1977). One possible explanation for this difference is that the states and operators of water jar puzzles are inherently easier to remember than the states and operators of river crossing puzzles. A second possible explanation is that the computer system used in the present study delivered the memory test more promptly, thus minimizing decay of the memory trace. Finally, the difference in performance level might only represent normal sampling variation. These hypotheses cannot be evaluated without further data.

Given that memory for problem solving steps is good but not perfect, it would be useful to know if some types of problem solving steps are remembered better than others. Table 2 shows memory accuracy and confidence judgments from Experiments 1 and 2 for four different types of steps. "Final step" is the action of indicating that the goal state has been reached, "moves on path" are pouring actions on the subject's path to the goal state, "moves off path" are pouring actions that lead away from that path; "corrections" include both backups and restarts. The same pattern is reproduced in both experiments and in both variables: Finishing actions are remembered almost perfectly; moves on path are remembered better than moves off path, which in turn are remembered better than correction steps. This pattern, if replicated in future studies, supports learning mechanisms which induce new problem solving rules from correct steps (e. g., composition, generalization) over mechanisms which derive new rules from incorrect steps and errors (e. g., constraint violation, discrimination).

Table 2. Memory accuracy and confidence judgements for steps that finish a problem, moves on path, moves off path, and corrections, for all subjects in Experiments 1 and 2. Confidence was rated on a five-point scale on which 5 indicated highest confidence.

	Type of problem solving step			
Experiment	Final step	Moves on path	Moves off path	Correction steps
Exp. 1: Path retracing				
Accuracy	.96	.76	.73	.43
Confidence	4.3	3.6	3.0	2.0
Exp. 2: Operator recall				
Accuracy	.97	.79	.56	.24
Confidence	4.7	3.3	3.0	2.0

It is possible that the subjects changed their behavior on the problem solving tasks so as to maximize their performance on the memory tests. For example, they might have rehearsed each problem solving step, something they probably would not normally do during problem solving. This possibility cannot be conclusively ruled out on the basis of the present data. But three aspects of the data argue against this possibility. First, the subjects learned to solve water jar problems; their problem solving performance improved across trials. Hence, they did not ignore the problem solving task. Second, if the subjects used various strategies for maximizing their memory performance, then we would expect their memory performance to improve across trials, but this was not the case. Third, although one can imagine that rehearsal of steps would help memory performance on the first three experiments, it is not obvious what kind of memory strategy one could use to improve performance on the order recognition test. But the subjects performed well on this test also.

It is also possible that the subjects ignored the instruction to respond to the memory items by trying to remember what they did during problem solving, and instead answered them by re-solving the problem. This possibility cannot be conclusively ruled out. However, re-solving the problem is obviously easier to do while retracing a solution path than while recalling operators in random order, which implies that the subjects should have performed significantly better in Experiment 1 than in Experiment 2. But this was not the case.

The results of this study support the notion that people store a detailed description of their own problem solving behavior while solving problems. The conclusion implies that it is *possible* that people learn new rules through induction over past problem solving steps. The present study does *not* address the question whether, in fact, there is a causal link between memory storage and procedural learning. If there is such a link, one would expect rate of procedural learning to be predictable from memory performance. But the data presented here cannot be used to investigate the correlation between memory performance and problem solving performance, for two reasons. First, it is not clear how to compute a measure of rate of learning from these data. Second, the uniformly high performance on the memory tests causes a restriction of range on the presumed predictor variable, precluding finding a significant correlation, even if it exists. Trying to predict rate of procedural learning from memory performance is a task for the future. The present study does not prove that our current theory of procedural learning is true, but it provides support for one of its basic assumptions, support which is independent of the learning curves that prompted the formulation of the theory.

Acknowledgements

Preparation of this manuscript was supported in part by grant no N00014-89-J-1681 from the Cognitive Science Program of the Office of Naval Research. Nancy Bee and Joyce Holl did accurate and reliable work on the data files.

References

Anderson, J. R. 1983. *The architecture of cognition*. Cambridge, MA: Harvard University Press.

Atwood, M. E., Masson, M. E. J., and Polson, P. G. 1980. Further explorations with a process model for water jug problems. *Memory & Cognition* 8:182-192.

Langley, P. 1987. A general theory of discrimination learning. In D. Klahr, P. Langley, and R. Neches, eds. *Production system models of learning and development* (pp. 99-161). Cambridge, MA: MIT Press.

Luchins, A. S. 1942. Mechanization in problem solving. The effect of Einstellung. *Psychological Monographs*, 54(6), Whole No. 248.

Newell, A. 1990. *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., and Rosenbloom, P. S. 1981. Mechanisms of skill acquisition and the law of practice. In J. R. Anderson, ed. *Cognitive skills and their acquisition* (pp. 1-55). Hillsdale, NJ: Erlbaum.

Newell, A., and Simon, H. A. 1972. *Human problem solving*. Englewood Cliffs, NJ: Prentice-Hall.

Ohlsson, S. 1987. Transfer of training in procedural learning: A matter of conjectures and refutations? In L. Bolc, ed. *Computational models of learning* (pp. 55-88). New York, NY: Springer-Verlag.

Ohlsson, S., & Rees, E. 1991. Adaptive search through constraint violation. *Journal of Experimental and Theoretical Artificial Intelligence* 3:33-42.

Reed, S. K., and Johnsen, J. A. 1977. Memory for problem solutions. In G. H. Bower, ed. *The psychology of learning and motivation. Advances in research and theory. Vol 11* (pp. 161-201). New York, NY: Academic Press.

VanLehn, K. 1990. *Mind bugs. On the origins of procedural misconceptions*. Cambridge, MA: Bradford Books.