

Tower of Hanoi: Evidence for the Cost of Goal Retrieval¹

John R. Anderson
Scott Douglass
Carnegie Mellon University

Address Correspondence to:

John R. Anderson
Department of Psychology – BH345D
Carnegie Mellon University
Pittsburgh, PA 15213-3890
Email: ja+@cmu.edu

Abstract

Past research on the Tower of Hanoi problem has provided clear evidence for the importance of goal-subgoal structures in problem solving. However, the nature of the traditional Tower of Hanoi problem makes it impossible to determine whether there is any special cost associated with storing or retrieving goals. A variation of the Tower of Hanoi problem is described that allows one to determine separately if there is an effect of how long a goal has to be retained on storage time or how long ago it was formed on retrieval time. This paradigm provides evidence for an effect of retention interval on retrieval time and not on storage time. An ACT-R (Anderson & Lebiere, 1998) simulation of these data is described that treats goal memory as no different from other memories.

The Tower of Hanoi problem has become a tool in cognitive psychology to understand planning both from an information-processing perspective (e.g., Simon, 1975) and a cognitive neuroscience perspective (e.g., Goel & Grafman, 1995). A simple version of this problem is illustrated in Figure 1. There are three pegs and four disks of differing sizes. The disks have holes in them, so they can be stacked on the pegs. The disks can be moved from any peg to any other peg. Only the top disk on a peg can be moved, and it can never be placed on a smaller disk. The disks all start out on peg A, but the goal is to move them all to peg C, one disk at a time, by means of transferring disks among pegs.

Until the 1970s the Tower of Hanoi was a puzzle of mild interest to mathematicians (e.g., Claus, 1884; Domoryad, 1944; Hinz, 1982) and just one of a stock of problems used by researchers interested in the effects of various manipulations on problem solving (e.g. Gagne & Smith, 1962; Ewert & Lambert, 1932). Then, reflecting the rise of the information-processing perspective, its structure became a task of special interest to psychologists (e.g., Egan & Greeno, 1974; Simon, 1975; Hayes & Simon, 1974). In Simon's 1975 paper he described a number of methods for solving the problem that required a goal-subgoal structure. One of these methods was the goal-recursion strategy which prescribed that, in order to move a pyramid of n disks from Peg A to Peg C, one moved a pyramid of $n-1$ disks from Peg A to Peg B, the n th (largest) disk to peg C, and then the pyramid of $n-1$ disks from peg B to Peg C. The strategy applied recursively to moving the smaller pyramids. He also described a variety of "sophisticated perceptual" strategies that basically prescribed focusing on the largest disk out of place, finding the largest disk blocking its move, and then focusing on getting that blocking disk out of the way. Thus, these sophisticated perceptual strategies also involve subgoaling. We will describe such a sophisticated perceptual strategy in detail later as it forms the basis of our experiment.

While all of these strategies required creating and retrieving subgoals, Simon also described strategies that avoided maintaining subgoals including a rote strategy of memorizing the move sequence.

It has been noted (Anderson, 1990, Kotovsky, Hayes, & Simon, 1985) that a natural human response to the Tower of Hanoi problem is to try to move disks to their ultimate destinations. This does not get one very far in solving the problem because disks have to be sometimes moved away from their ultimate destination in order to let larger disks get to their destination. To the degree that the participant is not aware of the dependencies among the disks (i.e. that movement of larger ones depends on movement of smaller ones) they will be reluctant to make such moves. The dependencies are relatively apparent in the original version of the problem but isomorphs (e.g., Kotovsky, Hayes, & Simon, 1985) have been produced that make these dependencies less apparent. These isomorphs tend to be more difficult and one reason is that participants find it harder to recognize the dependency structure of the problem and persist longer with ineffective strategies.

While naive participants do not approach the tower problem with a goal-subgoal strategy in place, they often discover it. Anzai and Simon (1979) report a detailed protocol of one participant discovering the pyramid strategy. Errors and latencies tend to be high in the problem at those points that require formulation of one or more subgoals. Anderson, Kushmerick, and Lebiere (1993) and Ruiz (1987) both report that time to make a move increases linearly with number of new subgoals that have to be created for that move in a sophisticated perceptual strategy. Given that there are strategies that do not require subgoaling, it is of interest that most participants come to adopt some sort of subgoaling strategy. However, we know of no study that has investigated how uniformly participants come to adopt a subgoaling strategy. Informally, the

first author discovered that one of his children had, from a computer version of the game, learned to employ what Simon calls the move-pattern strategy that does not involve subgoal. It seems fair to believe that there are exceptions to the tendency to subgoal.

Much of the post-70s research on the Tower of Hanoi has focused on how people discover the subgoal strategy or other learning trends (e.g., Anzai & Simon, 1979; Cohen & Corkin, 1981; van Lehn, 1991), what factors make it harder to discover the strategy in some isomorphs (e.g., Hayes & Simon, 1974, 1977, Kotovsky, Hayes & Simon, 1985; Simon & Hayes, 1976), and how easy it is to transfer this strategy from one isomorph to another (Kotovsky & Fallside, 1988). This paper will, however, focus on the execution of a particular subgoal strategy. Our interest is not particularly in this strategy or the Tower of Hanoi, but rather what the execution of this strategy says about how people process goals and subgoals generally. We will show that standard versions of the Tower of Hanoi task make it difficult to draw experimental inferences about how participants process the subgoal structure of the task and we will introduce a new version of the task that avoids past problems. The real focus of this paper is to understand how people process goals and subgoals. The research on the Tower of Hanoi problem indicates that humans have a strong proclivity to evolve a goal-subgoal strategy in this problem, but the research has not elucidated much about the memory mechanisms that support such a strategy.

The Tower of Hanoi is not the only task where we find evidence that subgoal is a natural human strategy. Studies of the somewhat similar Tower of London (e.g., Shallice, 1982; Ward & Allport, 1997) reveal subgoal. This tendency to subgoal has been found by Klahr and Robinson (1981) in children's solving of a number of puzzles. It appears in numerous academic problem-solving activities such as mathematical problem solving (Catrambone, 1995)

and programming (Anderson, Farrell, & Sauers, 1984). It is a basic mechanism in problems studied in Newell and Simon (1972) and indeed the phrase "means-end problem-solving" comes from the general observation that humans sometimes temporarily adopt the means as their end. Anderson (2000) has argued that a subgoaling capability is an important intellectual prerequisite for tool building.

While subgoaling is a natural human response to the recognition of dependencies in a problem, it is another question how much of human cognition involves subgoaling. Again, we know of no formal analysis but it strikes us that human cognition is often devoid of subgoaling and simply involves responding to the current situation. This was the argument of Suchman (1987) in her critique of planning theories in cognition. For the current purposes of the paper, it is enough to establish that subgoaling is a cognitive activity that people sometimes do naturally without trying to establish exactly what proportion of everyday cognition involves subgoaling.

Our interest in subgoaling and the Tower of Hanoi is partially motivated by our work with the ACT-R production system architecture (Anderson & Lebiere, 1998). One interesting commonality of production-system architectures (Anderson, John, Just, Carpenter, Kieras, & Meyer, 1995) is that they all have basic mechanisms to support subgoaling. This probably reflects their concern with problem-solving tasks, particularly use of artifacts like computer systems where such subgoaling is apparent. These architectures include the postulation of a goal stack that keeps track of goals. The issue of the goal stack has come in for some intensive analysis within the ACT-R architecture (Altmann & Trafton, 1999; Anderson & Lebiere, 1998) and the Soar architecture (John, 1996; Young & Lewis, 1999) with some people (e.g., Altmann & Trafton) viewing it as too powerful. We will illustrate this issue with respect to our own ACT-R system.

ACT-R's goal stack is a last-in-first-out (LIFO) goal stack. Such a stack naturally supports the hierarchical goal decomposition in the Tower of Hanoi since the larger disks will be represented by older goals on the stack and subgoals to move smaller disks will be represented by more recent goals. When a subgoal of the current goal is created that subgoal is said to be "pushed" on the stack and becomes the new focus of problem solving. Pushing really involves two things. First, the new subgoal is created and made the focus of attention. Second, an intention is set to refocus on the parent goal when the subgoal is achieved. When a goal is achieved it is said to be "popped". Popping a goal similarly involves two things. First, it involves retrieving the parent of the current goal and, second, attention is focused back on that parent goal. ACT-R's goal stack (as is true of SOAR) displays perfect memory in that the stack can retain arbitrarily many goals and they will be perfectly retrieved. It takes no time either to store a goal or to retrieve a goal. It seems plausible that storage might be effortful and take time and that retrieval might be fallible and also take time. As Anderson and Lebiere (1998) argued, the perfect goal memory assumption was made part of ACT-R as a simplification until data could be gathered to indicate the nature of the goal memory in humans. While the research in this paper will show that the perfect goal assumption is wrong, it probably required no formal research to establish this. The real question concerns the nature of the costs associated with goal memory. One purpose of our research will be to identify and separate the cost of creating and storing a subgoal from the costs of retrieving the subgoal. This will enable us to explore the question of whether these memory costs are similar to the costs people display for other kinds of memories they create and retrieve. To announce our main conclusion up front, goal memory does not seem different from other memories. Therefore, it does not seem necessary to assume a separate goal stack, perfect or fallible.

A Sophisticated Perceptual Strategy

Any Tower of Hanoi problem can be solved by applying the following strategy which is close to the sophisticated perceptual strategy described by Simon (1975):

1. Find the largest disk not in its goal position and make the goal to get it in that position. This is the initial “goal move” for purposes of the next two steps. If all disks are in their goal positions, the problem is solved

2. If there are any disks blocking the goal move, find the largest blocking disk (either on top of the disk to be moved or at the destination peg) and make the new goal move to move this blocking disk to the other peg (i.e., the peg that is neither the source nor destination of this disk). The previous goal move is stored as the parent goal of the new goal move. Repeat this step with the new goal move.

3. If there are no disks blocking the goal move perform the goal move and

- (a) If the goal move had a parent goal retrieve that parent goal, make it the goal move, and go back to step 2.
- (b) If the goal had no parent goal, go back to step 1.

This strategy is guaranteed to solve all Tower of Hanoi problems in contrast to Simon’s goal recursion strategy that will only solve problems that involve moving a tower from one peg to another.² However, the sophisticated perceptual strategy will not always find the shortest solution in the case where one is moving between two non-tower configurations (Hinz, 1992). It also has a resilience in the presence of goal forgetting. Should one forget subgoals at any point one can simply start over and calculate the goal structure. Actually one does not have to remember the subgoals at all since each subsequent move can be calculated by comparing current state to goal state and reapplying the strategy. However, this would be quite inefficient

and, as we will see, participants do retain subgoal information from one move to another, largely avoiding the need to recalculate.

Table 1 presents an analysis of the how the sophisticated perceptual strategy would apply to solving a 4-disk tower-to-tower problem in Figure 1. Part (a) of the table lists the steps required to solve the problem including the initial encoding of the problem and the goals that have to be formulated. It also lists which goals need to be stored for future action and which goals can be acted upon. In the case of goals that need to be stored for further action we have also indicated when they are retrieved. A particular goal might be retrieved multiple times before it can finally be acted upon. For instance, the goal of moving the 4 disk is retrieved three times before it can be finally executed. We have assumed that there is an initial encoding of the problem at the beginning. This involves noting where the disks are in the start and goal states. As a somewhat unrealistic approximation we assume perfect memory after this initial encoding of the problem and no need for any re-encoding.

Table 1b presents an analysis of the number of problem encodings, goal formulations, goal storages, and goal retrievals required for each of the 15 moves involved in performing the problem. Also included in the table is average move times for 28 participants in Anderson, Kushmerick, and Lebiere (1993) on those occasions where they executed a perfect 15-step solution with no errors. Table 1b also displays the intercorrelations among the variables. From an experimental design perspective, the problem is that these independent variables of encode, formulate, store, and retrieve are strongly intercorrelated. The first move in Table 1b might be discarded because it is the only one that involves encoding and so might distort the correlations. The correlations among the remaining variables are also displayed in Table 1b. As can be seen, excluding the first move does little to change the high correlations.

The experiment and data in Table 1b are typical of data on the Tower of Hanoi. The data show a clear effect of goal structure since there are long latencies at each point where goals are being formulated. However, the data provide no information beyond the fact that participants are formulating goals. In particular, it does not allow us to determine how much cost is associated with storing these goals or retrieving them—issues relevant to understanding how goals are processed. There is a strong correlation between number of goals that need to be stored and processing time but goal storage and goal formulation are so strongly correlated as to be indistinguishable. Number of retrievals has a negative correlation with time, which might seem to rule out goal retrieval as a significant variable, but retrievals have a strong negative correlation with goal formulation. Thus, all that the traditional form of the Tower of Hanoi tells us is that there is a strong effect of goal structure and hence that participants are using goals. It does not allow us to identify the costs of goal storage or retrieval. The purpose of the experiment to be reported here is to redesign the Tower of Hanoi task to enable us to identify what effects there are of goal storage and of goal retrieval.

Altman and Trafton (1999) have proposed that participants engage in a process of rehearsing goals in proportion to how long they will have to retain them before retrieval. This would predict an effect of how long the item must be retained on how long it is initially processed. Another problem with the design in Table 1b is that multiple goals are stored before a move and it is not possible to track the rehearsal differences that different goals might be receiving. On the other hand, if participants did not give special rehearsal to goals one might expect that they would be slowed in retrieval in proportion to how long ago the goals were created. Our more specific purpose in this experiment will be to separate the proposal that storage time will be sensitive to the future retention interval from the proposal that retrieval time

will be sensitive to the past retention interval. This is the sort of information needed to judge different models of goal processing.

Experiment

The purpose of the experiment was to use a variant of the Tower of Hanoi problem that would allow us to assess the cognitive costs in implementing a subgoaling strategy. There were three changes that we made to the way a Tower of Hanoi experiment is traditionally performed. The first involved resolving the design problem illustrated in part (b) of Table 1 and this resolution is illustrated in part (c) of the table. Basically, we had the participant perform a separate action for each subgoal formulated and so we no longer have the problem of multiple cognitive steps mapping onto a single action. That is, they were required to indicate the subgoals involved in the strategy as well as the moves. Figure 2 shows the interface that we used. There is a problem configuration, a representation of the ultimate goal (final state in Figure 2), a hidden goal stack onto which goals are posted, buttons for posting or doing goals, and a window for error messages. Participants post goals and make moves by selecting a disk and destination and either clicking the **Post Goal** or **Do It** button. They are required to post all goals that they cannot act on immediately. In posting the goals participants have to go through the same sequence of actions as they do when making that move to achieve that goal (i.e., selecting a disk, a destination peg, and clicking an action button). Thus, whenever participants formulate a goal they must either post that goal or make that move. This gives us one action per goal formulation and does away with the hidden goal formulations that contribute to the difficulty of interpreting the data. As can be seen from Table 1c this both gives us more observations and reduces the intercorrelations among the variables. In making the goal structure more explicit, we may be causing participants to use subgoaling more than they normally would. This is a desirable

outcome because our intention is to study the mechanisms that support subgoalting. We are not interested in the Tower-of-Hanoi problem per se nor documenting the mixture of strategies that participants might bring to this problem.

In Table 1c we have also put the number of intervening actions over which goals have to be retained in the **Store** column. Thus for instance, the Post 3 (Action Number P2) has to be retained over the post 2 action, move 1 action, and move 2 action before it has to be retrieved again to figure out where to move the 1. Hence, its future retention interval is 3 actions.

Similarly, in the **Retrieve** column we have placed how many actions a goal has been retained before it is moved. Table 1c also presents data from the experiment to be reported here for the subset of problems that had an equivalent a goal structure to this particular tower problem. It is clear from these data that the first action of posting the 4 goal is special and takes much longer. This is because this is the move on which the participant must encode the problem. Note that there is still a strong correlation between encoding and goal storage in this problem. Most of our analyses will be focused on the 10-action subsequences that involve moving the 3 towers. There are two of these in Table 1 c (Actions 2 - 11 and Actions 13 - 22). These action sequences avoid the confounding with initial encoding since that has occurred before the beginning of such sequences.

There are two other factors that frustrate interpreting data from a typical Tower of Hanoi experiment. Our solutions to these two problems are related. One problem, that we have discussed earlier, is that participants do not always immediately apply a subgoalting strategy to the task but rather tend to discover it with practice. Our interest is not in strategy discovery but execution of a particular subgoalting strategy. Therefore, we instructed participants on the sophisticated perceptual strategy and gave them a practice problem to master the strategy before

collecting data for the experiment. However, this practice problem exacerbates another difficulty, which is practice effects. If one repeats the same problem over and over again participants come to memorize the steps to solve this problem (Ruiz, 1987). However, one does not have to repeat the same tower-to-tower problem. Figure 3 shows some 4-disk problems that are all 15 moves apart and involve the same subgoal (7 subgoals posted as in Table 1c) but these problems have the disks in different configurations. There are a great many equivalent problems³ and we do not have to repeat problems. In this experiment we will have participants solve 21 problems for each of 3 days (after their warm-up problem on Day 1). Not only did these problems vary in their surface appearance but, as we will discuss in the method section, they also differed in the number of moves. This also allows us to obtain replication without losing effects of goal structure to memorization and so increases our accuracy of measurement. Accuracy of measurement has been a problem in many Tower of Hanoi experiments where each participant only gets to solve a single problem before serious learning effects set in and change the structure of the problem solving. Accuracy of measurement becomes key if we want to assess more subtle effects associated with subgoaling than merely to demonstrate the existence of subgoaling.

Method

Participants and Instructions.

Ten participants were recruited from the Carnegie Mellon University student population to participate in a 3-day experiment. While it is difficult to find participants at Carnegie Mellon who have not seen the Tower of Hanoi, none of the participants had much experience with it and all found the problems challenging. Only participants with normal uncorrected vision (no glasses or contact lenses) were allowed to participate in the experiment. Participants completing

all three experimental sessions were paid a base fee of \$5 and a performance bonus of \$0.002 (1/5 of a cent) per trial point accumulated over all sessions. Trial points were determined by subtracting trial completion time (in seconds) from 10 times the number of actions required to solve the trial's specific puzzle. In addition to describing the Tower of Hanoi and the computer systems, participants received the following strategy and motivational instructions:

“There is a great variety of moves which can be made in solving this puzzle. In this experiment, you will be constrained to making only correct moves under a goal-oriented, recursive strategy. That strategy is laid out as follows:

1. Formulate a Goal. You should determine which disk is the largest out of place (not in its goal state). Formulate a goal to move a disk to its goal peg by clicking the disk, then the destination peg.

2. Click a Button. If you can make a legal move to achieve the goal you have posted, choose “Do It!” and skip to step 4. Otherwise, choose “Post Goal”, and your goal will be posted on the goal stack.

3. Formulate a Prerequisite Goal. Suppose you cannot move a disk (we'll call it D) to a particular peg (call it P). Find the largest disk that is blocking its move (a blocker is either sitting on top of D, or is a smaller disk than D occupying P) and formulate a goal to move it out of the way. The peg to which you should move this blocking disk is the peg which is not one of the following:

- (a) The peg that D is currently on.
- (b) The peg P to which you want to move D.

4. Try Again. Go back to step 2 to see if you can achieve the last goal you posted.

5. Repeat the Process. Go back to step 1 and continue until all disks are on their goal pegs.

You start each puzzle with a certain number of points; these points tick away with the amount of time it takes you to complete the task. You should try to maximize your score by completing the puzzle quickly without making mistakes. If you do make a mistake, you'll hear a beep and see an error message below the goal line. Reading that message will help you correct your move to fit the strategy."

We judged that participants only partially understood what these instructions meant and they made numerous errors working out the strategy on their practice problem. However, after that they were able to maintain low error rates and a number of participants remarked they liked the strategy ("after I got the hang of it," as one participant added).

Problems. All the 15-move problems in Figure 3 correspond to the structure in Table 1—they involve performing 7 moves to enable the 4 disk to be moved to its destination, the move of the 4 disk, and 7 more moves to get the rest of the disks in place. The first 7 moves creates a tower of disks on the peg that is neither the source nor the destination of the 4 disk. The last 7 moves disassembles the tower into the goal configuration. If the 15 moves were a classic tower to tower problem, the first 7 moves would move the 3 tower from off the 4 disk to the other peg and the last 7 moves would move that tower on top of the 4 disk on the destination peg. To increase the variety of problems we also used 6 other problem types for a total of seven problem types. All problems actually had 5 disks, as in Figure 2, and four of the problem types involved a move of the 5 disk. For the other three problem types the 5 disk started out in its goal configuration and the 4 disk was the largest disk that had to be moved. These problems can be characterized by size of pyramid assembled, the largest disk then moved, and by the size of the

pyramid then disassembled. According to this classification the 15 move problems in Figure 3 were all 3-4-3 problems to reflect that they create a 3 tower, move a 4 disk, and disassemble the 3 tower. Each of these 15 move problems involve posting 7 subgoals as in Table 1. The full set of problem types were 4-5-4 (31 moves, 15 subgoals posted), 4-5-3 (23 moves, 11 subgoals posted), 3-5-4 (23 moves, 11 subgoals posted), 3-5-3 (15 moves, 7 subgoals posted), 3-4-3 (15 moves, 7 subgoals posted), 3-4-2 (11 moves, 5 subgoals posted), and 2-4-3 (11 moves, 5 subgoals posted).

The problem analyzed in Table 1 is a 3-4-3 problem. For ease of understanding it is described as a tower-to-tower problem but we used randomly generated configurations with the same logical goal structure. One of the purposes of this variety of problems was to prevent the participant from being able to predict exactly which disks would have to be moved or posted. A complete listing of all the moves and subgoals posting for each of these 7 problem types is available by following the Published Models link from the ACT-R home page:

<http://act.psy.cmu.edu/>. This source also contains a more complete breakdown of the data than will be reported here. There were three replications of these 7 problem types each day. The problems for each day and the practice problem were randomly generated under the constraint that there be no problem repeats.

Procedure. Figure 2 illustrates the computer display used with the participants. Overlaid on the display is a grid (that participants did not see) of lines that are separated by approximately one degree of visual angle. Each move or posting required selecting a disk, a destination peg, and either the do-it or post action. These actions were performed with a mouse. If a goal was posted it would appear on top of the stack of goals. The goals on the stack are hidden but participants could see an item of a stack by clicking on it. This gave us a way of assessing

whether participants actually forgot a goal. Of course, if they forgot a goal, they could reconstruct it by referring back to the goal configurations and reapplying the sophisticated perceptual strategy. We also collected eye movements because we thought that they would reflect such goal reconstructions. Participants were allowed to retract a disk or peg before they hit an action button if they changed their minds. We developed and used the following classification of participants' moves and posts according to their clicking behavior:

- Goal inspection: participant clicks on goal element in the goal stack to be reminded of it.
- Disk error: participant selects wrong disk to move or post.
- Peg error: participant selects the right disk but moves it to the wrong peg.
- Action error: participant selects the right disk and peg but the wrong action (post versus do it).
- Disk slip: participant initially selects the wrong disk but self-corrects.
- Peg slip: participant initially selects the wrong peg but self-corrects.
- Perfect: participant makes a perfect sequence of disk, peg, and action selections with no slips nor any inspection of the goal.

These categories were made mutually exclusive in that a move or post was classified in a lower category only if it did not satisfy any higher category. If the participant made an error a feedback message was given to the participant explaining the nature of the error and the participant had to try again. We did not analyze the repeated efforts. The latency measures we will present are only for perfect trials and reflect the total time for the three mouse clicks (disk, peg, action).

While participants performed this experiment their eye movements were monitored. Participants were calibrated at the beginning of the day and were recalibrated if necessary after

solving a particular problem. The eye tracker used to determine participant point-of-regard (POR) during the trials is an ETL-500 manufactured by ISCAN, Inc. The ETL-500 is a high-resolution pupil/corneal reflection dark pupil tracker that uses a Polhemus FASTRAK magnetic head tracker to compensate for participant head movements during tracking episodes. Under typical operating conditions, the ETL-500 estimates participant POR with an accuracy better than one degree of visual angle. The tracker was configured with ultra-light head-mounted optics to minimize participant discomfort. Trial event, mouse position, and participant POR data were recorded every 16.7 ms by the experiment delivery software. By interleaving task state information with the participant POR data, the experiment delivery software was able to record trial data files that could be analyzed and replayed after the experiment.

Design and Analysis. The seven problem types involved a subsequence of 10 actions that occurred 1 to 4 times depending on the problem type. This sequence of actions was logically equivalent to solving a 3-tower problem although often it did not involve a tower. This sequence was

Post the 3 disk
Post the 2 disk
Move the 1 disk
Move the 2 disk
Move the 1 disk
Move the 3 disk
Post the 2 disk
Move the 1 disk
Move the 2 disk

Move the 1 disk

This sequence of 10 actions will serve as a major organizing factor in our analysis of the data. As we will explain below, these sequences of 10 actions can be aggregated across problems into three categories depending on the logical structure of where they occur. This allows for a 3x10 factorial analysis of the data. These three categories are:

(a) End. Six of the seven problem types (the exception being the 3-4-2 problems) ended with this sequence. This ending sequence of moves starts with a 3-tower and the subject's actions disassemble the disks from this tower into their final configuration. It can be viewed as a mini-problem. We call these "End Sequences." The average latencies associated with the ending sequences for the six problems have an average intercorrelation of .57⁴ and the percentage of perfect moves correlate .83.

(b) Under 4. Another common context in which this sequence of 10 moves appears is just after posting a goal to move a 4 disk and serves the purpose of moving the 3 smaller disks out of the way of the 4-disk so that it then can be moved. This occurs twice in the 4-5-4 problems, once in each of the 4-5-3 problems, the 3-5-4 problems, the 3-4-3 problems, the 3-4-2 problems. We call these "Under 4" because they are all performed when the goal to move the 4 disk is on the stack. The latencies across these 6 patterns correlates an average of .78 and the accuracies correlate .60.

(c) Under 5. There are four remaining instances of these ten-action sequences that served the purpose of moving the 3 smaller disks out of the way of the 5-disk. Two, occurring in 3-5-4 and 3-5-3, involve posting the 3 goal immediately after posting the 5 goal. Two, occurring in 4-5-4 and 4-5-3, involve posting the 3 subgoal after the 4 disk has been moved and while a goal to move the 5 disk is still active. We call these four sequences "Under 5" because they are all

performed when the goal to move the 5 disk is the only goal on the stack. Their latencies correlate an average .89 and their accuracies correlate an average of .88.

Results

Main Analysis The factors for the main analysis were category (End, Under 4, Under 5), days (3 values) and actions (10 values). For each cell in the 3 x 3 x 10 design, we calculated for each participant a mean latency (with a maximum of 5 seconds to avoid outliers) and the percent perfect moves. The latencies averaged over participants and days are displayed in Figure 4 and the accuracies in Figure 5. The main effect of categories was significant ($F(2,18) = 10.96$, $MSE = 0.44$, $p < .001$ for latency; $F(2,18) = 6.46$, $MSE = 0.010$, $p < .01$ for accuracy) with participants best on the End Sequences (2.75 sec per action, .932 perfect), next best on Under 4 (2.89 sec, .930 perfect), and worst on Under 5 (3.00 sec, .906 perfect). The main effect of actions was also significant ($F(9,81) = 5.67$, $MSE = 1.15$, $p < .0001$ for latency; $F(9,81) = 12.35$, $MSE = 0.031$, $p < .0001$ for accuracy) as was the interaction of actions with conditions ($F(18,162) = 6.91$, $MSE = 0.25$, $p < .0001$ for latency; $F(18,162) = 9.44$, $MSE = .009$, $p < .0001$ for accuracy). The effect of days was also significant ($F(2,18) = 9.81$, $MSE = 2.69$, $p < .01$ for latency; $F(2,18) = 5.50$, $MSE = 0.021$, $p < .05$ for accuracy) with participants getting better as a function of days—3.20 seconds and .901 perfect on Day 1, 2.81 seconds and .926 perfect on Day 2, and 2.62 seconds and .941 perfect on Day 3. While some of the interactions with days are significant, the nature of the interactions is just to reduce the size of the effects approximately proportionately. The average day-to-day intercorrelation over the 30 conditions is .92 for latency and .88 for accuracy.

As we will see, the model we will present will predict that there is constant accuracy and latency for all positions except action 1 (push 3), action 5 (move 1), and action 7 (push 2).

Removing these three actions leaves us with a 3 x 7 category x action design that has no significant main effects of action, or category, nor a significant interaction between the two.

Thus, we have 21 cells that we can treat as a baseline. We calculated t-tests to see which of the remaining 9 cells were significantly different than the average of mean of these 21 cells, using the overall interaction term between the 30 conditions and 10 participants to get an estimate of variance. For latency, the non-significant cells were the action 1 for the Under 4 category and End categories, action 5 for the End category, and action 7 for the End category. For accuracy the only cell that failed to be significantly different at the 0.05 level was the action 1 for Under 4 category.

Participants find the action 1, the posting of the 3-disk, in the Under 5 category the most difficult. We think there are two different reasons for this difficulty corresponding to the two subcases in this category. There are the cases where the participant has just posted the 5-disk before posting the 3-disk. In this case, they first have to consider whether the 4-disk needs to be moved, determine that it does not, and then focus on the 3-disk. The greater difficulty of this step reflects this extra intervening processing. There are the cases where they have just moved the 4-disk and in this case the difficulty involves retrieving their intention for the 5-disk so they can determine where to move the 3-disk. It was 12 actions earlier that they had set their intention for the 5-disk. Consistent with this analysis is the fact that participants request seeing the goal 7% of the time for the action 1 in the Under 5 category when it occurs after a move of the 4-disk but only 2% when it occurs after a push of the 5-disk. A similar retrieval explanation accounts for the difficulty of the action 7 in the Under 4 and Under 5 categories. In both of these cases they have to retrieve the intention for the 4-disk or the 5-disk that they set 6 actions earlier just before this sequence of 10, in order to decide where to post the move for the 2-disk. In

contrast, for action 7 in the End sequence there is no goal on the stack and participants can formulate their posting for the 2-disk by comparing the goal and current state. Consistent with this explanation participants request to see the goal stack 5% of the time for action 7 in the Under 5 category and 8% of the time in the Under 4 category. In contrast, they never make this request at action 7 in the End category. Over all other actions, they average less than 1% requests to see the goal stack. Note that these cases where the goal stack is inspected are excluded in our latency calculations. Thus, the increased latencies plotted in Figure 3 reflect either time to retrieve the goal or time to reconstruct it.

This retrieval problem may also be part of the difficulty for action 5 (moving the 1-disk). To decide where to move this disk, the participant must retrieve the goal they posted for the 3-disk three actions earlier. However, there is only a slightly elevated rate of requesting to look at the goal stack (2%). Another unique feature of the data involving this move is that participants often first select to move the 3-disk and then change their mind and choose the 1-disk. Such disk slips occur on 9% of the trials for this move compared to 2% for all other actions. Apparently, participants retrieve their goal to move the 3-disk and fail to notice initially that its move is blocked. While these overt disk slips are excluded in our latency analysis they may be occurring implicitly as well, contributing to longer latencies on perfect moves.

Eye Movements We did an analysis of where participants fixated while they performed these sequences of 10 actions. Figure 6 shows the regions we used. Fixations were identified as periods of low velocity in eye movement. To attribute these fixations to regions of interest (ROIs) on the task screen (to determine where participants were looking during various portions of the task) fixation centroids were used to determine which, if any, of the ROIs in Figure 6 the fixations occurred within. Fixations falling outside any of these ROIs were attributed to a

middle-of-nowhere (MON) region. We calculated for each participant the average duration of all fixations in a particular region during each by action x category x day condition.

We aggregated the ROIs into three areas of interest:

(a) Action Areas: This includes the peg with the disk being acted upon, the peg that is the destination, and the portion of the screen that contains the post and do-it buttons. Participants average 84% of their time fixating these regions as they guide their mouse movements. The sequence of eye movements is typically to the disk, the peg, and then the action.

(b) Planning Areas: In Figure 6 these are the final state and the other peg in the current configuration. There is no reason to be looking at these regions unless one is planning a move and cannot remember information for this planning. Participants also need to fixate these areas to initially encode the problem but their initial moves are excluded from the current analysis that is restricted to these sequences of 10 actions that occur after initial encoding. Participants average 4% of their time fixating these two regions during the moves in this analysis.

(c) Other Areas: These are the rest of the task screen and off screen. Participants spend 12% of their time fixating these regions.

Relatively little time is spent fixating planning regions. It is only necessary to fixate planning regions when one is both planning and has forgotten the goal or current state. So this low frequency of fixation indicates that participants largely were able to keep the problem and the goal configuration in mind. (As another sign that participants were able to keep the goal in mind, they only requested to inspect the goal stack 1% of the time.) Still this 4% of the fixations is highly informative because it does indicate when participants are planning and have forgotten critical information. Therefore, we decided to do an analysis of how the proportion of planning time varied as a function of the factors of the experiment. Figure 7 displays how the proportions

varied for the three categories as a function of action. There were significant effects of all three main factors -- actions ($F(9,81) = 30.64$; $MSE = .0021$; $p < .0001$), categories ($F(2,18) = 10.90$; $MSE = .0016$; $p < .001$ with participants spending 5.0% planning fixations for Under 5, 4.4% for Under 4, and 3.5% for End), and Days ($F(2,18) = 50.65$; $MSE = .0018$; $p < .0001$ with participants spending 6.3% planning fixations for Day 1, 3.8% for day 2 and 2.8% for Day 3). It should be noted this reduction across days, being a proportion, is over and above the overall reduction in total time. The two-way interaction displayed in Figure 7 between category and action is also highly significant ($F(18,81) = 10.64$, $MSE = .008$, $p < .0001$). Participants are showing significant rises in their proportion planning time at those points where they must retrieve their plans -- first action for Under 5, fifth action for all three categories, and seventh action for Under 4 and Under 5. So this provides further evidence that at these points participants are sometimes forgetting their goals and having to replan them.

Steps involving 4 and 5 Disks We also did a separate analysis of the posts and moves involving the 4 and 5 disks—these steps occur outside the 10 step sequences we have been analyzing so far. Table 2 presents these data classified by the size of the disk involved and whether it involves a post or a move. In the case of posts of disk 4 we distinguished whether it was the initial action in solving the problem (Problems 3-4-3, 3-4-2, or 4-3-2) or whether it was an internal action within the sequence (Problems 4-5-4, 4-5-3, or 3-5-4). There were main effects of both days ($F(2,18) = 18.74$, $MSE = 0.43$ $p < .0001$ for latency; $F(2,18) = 5.05$, $MSE = .021$, $p < .05$ for accuracy) and category in Table 2 ($F(4,36) = 160.69$, $MSE = 0.24$, $p < .0001$, for latency; $F(4,36) = 4.90$, $MSE = .016$, $p < .01$ for accuracy). The interaction between days and categories is significant for latency ($F(8,72) = 2.84$, $MSE = 0.14$, $p < .01$) but not accuracy ($F(8,72) = 1.29$, $MSE = .009$). The latency interaction is just that days of practice compress the

differences among categories. The average correlation among categories across days is .98. The big effect in Table 2 is between the two post conditions involving initial actions and the other three conditions involving internal actions—the contrasts for this are highly significant ($t(36) = 25.08$ for latency and $t(36) = 3.51$ for accuracy). The residual within category variance is significant only for latency ($F(3,36) = 4.62$, $p < .01$) and not for accuracy ($F(3,36) = 2.42$). Table 2 reinforces the conclusion that the actions involving disks 4 and 5 are special, particularly when they are the first action in solving a problem. The difference in Table 2 between posting a goal as an initial action and posting it as an internal action provides an estimate of how long it takes to encode the initial problem and that estimate is 1.85 sec.

Regression Analyses Table 3 presents an analysis of the 30 actions in Figure 4 in terms of the relationships among the variables of goal formulation, goal storage, goal retrieval, and latency. These actions have the advantage that none of them involve initial encoding and so we do not have to consider complications due to that. Recall that the first move in the Under 5 category merges two situations: (a) the participant has just set a goal to move the 5 disk, has determined the 4 disk is in the right position to enable this move, and is now subgoaling the 3 disk to enable the move of the 5 disk; and (b) the participant has just moved the 4 disk and now must retrieve the intention for the 5 disk which was set 12 actions ago. Since we are interested in the effect of this delay, for this table we only include the cases of type (b). The average of the cases where this move involves a retrieval of the 5 goal is slightly higher (3816 ms) than the mean for all cases in Figure 4 (3789 ms.). A linear combination of all three variables correlates fairly well with the data (multiple $R^2 = .900$) and the best fitting linear equation is

$$\text{Time} = 2.655 + .065 * \text{Formulate} + .053 * \text{Store} + .085 * \text{Retrieve}$$

If the Formulate variable is removed from the equation the multiple R^2 drops to .889; if the Store variable is removed the multiple R^2 drops to .873, and if the Retrieve variable is removed it drops to .315. It seems pretty clear that the Retrieve variable is the most critical. An equation involving just the Retrieve variable fits quite well with a multiple R^2 of .846. While the retrieve variable captures most of the variance, either of the other variables produces a significant increment to the variance explained. The best fitting equation involving just the retrieve variable is

$$\text{Time} = 2.615 + .094 * \text{Retrieve}$$

It is worth noting that none of these conclusions depend on the extreme values of 6 and 12 for the Retrieve variable. The results about the relative importance of the variables are the same with these cases excluded.

In conclusion, the latency data indicate that participants are slower at those points where they must retrieve goals and they are more slowed the longer ago it was that they formulated the goal. The accuracy data show a similar pattern suggesting participants are forgetting their goals. Moreover, their tendency to inspect the goal stack or look at the goal configuration increases dramatically at these retrieval points. Thus, goal retrieval seems to be the major factor limiting performance in this task.

An ACT-R Model

The ACT-R architecture (Anderson & Lebiere, 1998) has a "perfect memory" goal stack on which all goals can be stored perfectly and from which goals can be accessed without any retrieval time cost. However, Anderson and Lebiere (1998; p. 459) remarked that this was probably a simplification for convenience until data could be produced to show goal limitations.

The data reported here do just that. However, the real agenda in this research was to explore the nature of the limitations on goal memory.

Altman and Trafton suggested that memory for goals might behave like any other memory and be subject to forgetting. Again our data support this. They proposed that participants would engage in a rehearsal process to avoid forgetting goals and rehearse more goals that needed longer retention periods. Our data do give weak evidence for this in that there was a significant effect of the Storage variable over and above the retrieve variable. However, the dominant effect was due to the Retrieve variable. Thus, it seems much more the case that participants do not choose to give extra rehearsal to goals that they will have to remember longer but rather choose to pay the price of forgetting. In part this may reflect the particular task where they can always reconstruct their goal -- either by querying the goal stack or by reapplying the sophisticated perceptual strategy.

We will describe a model here that simulates the experimental data assuming no effort at rehearsal but rather only a forgetting process. The model does not use the goal stack in ACT-R but rather relies on ACT-R's general declarative memory to try to store and retrieve goals. Thus, this model-fitting enterprise serves as a test of the hypothesis that goals are stored and retrieved in the same system as other memories. A running version of this model can be obtained by going to the Published Models link at the ACT-R home page: <http://act.psy.cmu.edu/>.

Figures 4 and 5 illustrate the fit of an ACT-R model to the data. As can be seen the model gives a good fit, correlating .942 with the latency data and .909 with the accuracy data. There is basically just one factor producing the effects in this model. This is forgetting of goals. Whenever a goal is forgotten we assumed a 50% probability that the model will consult the goal stack, which is counted as an error, and a 50% probability that the model will try to reconstruct

the goal which we estimate took 1.85 seconds (the average difference between initial posting and internal postings in Table 2). Thus, the accuracy dips and latency spikes are two sides of the same coin and indeed the two dependent measures are highly correlated ($r = -.915$). Thus, the key to understanding the predictions of the model is to understand how probability of a memory failure varies with delay in ACT-R.

In ACT-R each chunk has a base-level activation that increases each time the chunk is used and decreases with lack of use. If a chunk has been used n times in the past at times t_1, t_2, \dots, t_n , ago the base-level activation is given by the following equation.

$$A = \ln \left(\sum_{j=1}^n t_j^{-d} \right) \quad \text{Base-Level Learning Equation}$$

In the equation, each presentation decays as a power function of delay with an exponent of d and each of these effects are then summed. Finally, these effects are squashed through a logarithmic function. The standard value of d for declarative memory in ACT-R is .5 and this was the value used in the simulation producing the fits in Figures 4 and 5.

Note that this formulates the effect of time somewhat different than the regression analysis where we are simply counting number of actions. Therefore, it is not a foregone conclusion that this analysis will be as successful as the regression analysis. The correlation of ACT-R with the latency data was .942, which implies an R^2 of .888 and this is better than the R^2 of .846 achieved in the regression analysis (Table 3) that just used the retrieval variable. This indicates that the formulation of the time variable in the Base-Level Learning Equation is, if anything, better than the formulation in the regression analysis.

The base-level activation of a chunk is noisy and will fluctuate around its expected value. Should the activation fall below a threshold, τ , it will fail to be recalled: The probability of this happening is described by the following equation:

$$\text{Probability} = \frac{1}{1 + e^{-(A-\tau)/s}} \quad \text{Retrieval Probability Equation}$$

where s is related to the standard deviation of the activation by the equation $s = \sqrt{3}\sigma/\pi$. The threshold τ was estimated to be -1.45 and s was estimated to be .15. To illustrate what these values mean, if a goal had been set 18 seconds earlier its base-level activation would be at the threshold and there would be a 50% chance of the goal being recalled. At a 13 second delay there would be a 75% probability of recall and at a delay of 25 seconds there would be a 25% probability of recall. With actions averaging just about 2.85 seconds this means that the 25th, 50th, and 75th percentiles are after approximately 5, 7, and 9 actions. Sometimes goals get additional rehearsals. For instance, the goal to move the 5 disk gets set initially, retrieved and rehearsed after the 4, 3, and 2-disks are moved out of the way. These extra rehearsals will increase the base-level activation according to the Base-Level Learning Equation.

Four parameters were estimated to fit the data; a τ of -1.45, an s of .15, a 2.7 second minimum time to execute the 3 mouse clicks associated with a move, and a 5% probability of some slip for any action producing an non-perfect action. As can be seen from Figure 4 the model predicts latency greater than the minimum time or errors greater than the slip rate on only a few actions, which are the ones where a goal has to be retrieved. In general, the data in Figures 4 and 5 correspond to the model's predictions. The one point where the model appears to fail to capture the data is the action 7 (push 2) for the End sequence. The model does not predict any retrieval errors at this position since the subgoal of moving the 2 disk is not calculated by

retrieval but rather by comparing the goal and the current state. In the Under-4 and Under-5 sequences, where errors are predicted for this action, the posting of the 2 disk is in service of another goal that has to be retrieved. Of the 16% errors at this position for the End Category, 7% involve trying to move the 1 disk rather than posting the 2 disk and 4% involve trying to move the 2 disk rather than posting it. Thus, it seems that errors at this position are due to premature moves intruding at the very end of the problem.

Conclusions

This experiment has provided clear evidence for subgoalings as have other experiments. However, it is the first experiment to elucidate the nature of subgoalings. In contrast to the Altmann and Trafton proposal, it does not appear to be the case that there is much cost in the Tower of Hanoi associated with creating a subgoal or storing a parent goal for later retrieval. On the other hand, in contrast to the predictions of the ACT-R and Soar architectures, the retrieval of stored goals is not certain and not without a time cost. If a goal was forgotten in the ACT-R model, there was a large cost of the 1850 msec to reconstruct the goal. As we noted, participants could have engaged in a rehearsal process to enhance their memory as indeed one can in any memory situation. However, they usually did not choose to, perhaps because they were pre-occupied solving the problem and did not focus on rehearsal.

The clear implication of this research is that cognitive architectures like ACT-R and Soar are wrong in their assumption of a special goal stack. Goals appear to behave like any other memory objects. Goals set in the process of subgoalings are probably no different than other sorts of intentions that people set. People set intentions all the time, such as to pick up food on the way home or to compliment a friend, and these intentions are not part of a highly organized dependency structure as in the Tower of Hanoi. It is of interest to compare memory for goals in

our task with research on what is called prospective memory (for a series of recent papers see Brandimonte, Einstein, & McDaniel, 1996). The typical prospective memory task is remembering to do something like returning a telephone call. Like our situation, success at such a prospective memory task can vary with the amount of effort put into initial encoding of the intention and the retention interval (which is often much larger than what we saw in this experiment). It is argued that prospective memory is distinguished from other memories by the fact that one not only needs to be able to successfully recall the intention but one needs to recall it in the appropriate context. It is noted that people can perfectly well remember what they are supposed to do when prompted but fail to prompt themselves. Memory failures in our Tower of Hanoi task are potentially ambiguous between whether the participants failed to prompt themselves to remember the goal or could not recall the goal when prompted. We think the latter interpretation is much more likely in our situation since both the completion of the current action and the existence of a non-empty goal stack on the screen should be reminders that there is a goal to remember.

While the perfect-memory goal stack is not supported, it remains plausible that subgoalting is in some sense cognitively primitive. As we noted earlier, Klahr and Robinson observed that even young children, without special instruction, form subgoals when they perceive the dependency structure in their environment. However, in our research we used sophisticated college students and especially instructed them on the subgoalting strategy that we wanted them to adopt. While this gave us the control needed to study the subgoalting process, it prevents us from making any conclusions about whether subgoalting is a cognitively privileged process. What we can conclude is that memory for subgoals is like memory for anything else and, in particular, that it shows a retention function similar to that of other memories.

It is worth considering how these conclusions might depend on the choice of the Tower of Hanoi problem and on the particular experimental methodology. We do not think the conclusion that goal memory is like other declarative memory is compromised by these choices. As we noted in the introduction, other tasks show similar subgoal effects and the current version of the Tower of Hanoi task was motivated to analyze these effects. However, it may be the case that the Tower of Hanoi task lends itself to the result that storage effects are weak and retrieval effects are large. Since it is easy to reconstruct the goal from the current display, the price of memory failure is not particularly high and there is little motivation to engage in goal rehearsal. However, we do not think the lack of rehearsal effects are in any way unique to our paradigm but in fact would be typical of display-based problem-solving tasks where the display allows goals to be reconstructed. Interestingly, Byrne and Bovair (1997) find that what they call post-completion errors (e.g., leaving a card in an ATM machine) occur when people forget a goal and the problem display does not support reconstructing the goal. Thus, people fail to adequately rehearse and suffer retrieval errors even in situations where the external display does not support goal memory. Nonetheless, our major conclusion is not that there are only retrieval effects on goal memory and not rehearsal effects. We would expect both rehearsal and retrieval effects in goal memory as in any other type of memory and, as in many memory tasks, rehearsal is a strategic decision on the person's part. Our major conclusion is that goal memory is like other more common kinds of declarative memory and will show the same effects of practice and retention interval.

References

- Altman, E. M. & Trafton, J. G. (1999). Memory for goals: An architectural perspective. In Proceedings of the 21st annual Conference of the Cognitive Science Society, pp. 19-24.
- Anderson, J. R. (2000). Learning and Memory, Second Edition. New York: Wiley.
- Anderson, J. R. (1990). The Adaptive Character of Thought. Hillsdale, NJ: Erlbaum.
- Anderson, J. R., Farrell, R., & Sauers, R. (1984). Learning to program in LISP. Cognitive Science, 8, 87-130.
- Anderson, J. R., John, B. E., Just, M. A., Carpenter, P. A., Kieras, D. E., & Meyer, D. E. (1995). Production system models of complex cognition. In Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society [pp. 9-12]. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Anderson, J. R., Kushmerick, N., & Lebiere, C. (1993). The Tower of Hanoi and goal structures. In J. R. Anderson (Ed.). Rules of the Mind. Hillsdale, NJ: Erlbaum.
- Anderson, J. R. & Lebiere, C. (1998). The atomic components of thought. Mahwah, NJ: Erlbaum.
- Anzai, Y, & Simon, H. A. (1979). The theory of learning by doing. Psychological Review, 86, 124-140.
- Brandimonte, M., Einstein, G. O., & McDaniel, M. A. (1996) (Eds.). Prospective memory: Theory and applications. Mahwah, N.J. : L. Erlbaum.
- Byrne, M. D. & Bovair, S. (1997). A working memory model of a common procedural error. Cognitive Science, 21, 31-62.
- Catrambone, R. (1995). Aiding subgoal learning: Effects on transfer. Journal of Educational Psychology, 87, 5-17.

- Claus, N. (1884). La Tour d'Hanoi, Jeu de calcul, Science et Nature, 1, 127-128.
- Cohen, N. & Corkin, S. (1981). The amnesiac patient H. M.: Learning and retention of a cognitive skill. Presented at the 11th Annual Meeting of the Society for Neuroscience, Los Angeles.
- Domoryad, A. P. (1964). Mathematical games and pastimes. Oxford: Pergamon Press.
- Egan, D. E. & Greeno, J. (1974). Theory of rule induction: Knowledge acquired in concept learning, serial pattern learning, and problem solving. In L. W. Gregg (Ed.), Knowledge and cognition (pp. 43-104). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Ewert, P. H. & Lambert, J. F. (1932). Part II: The effect of verbal instructions upon the formation of a concept. Journal of General Psychology, 6, 400-413.
- Gagne, R. M. & Smith, E. C. (1962). A study of the effects of verbalization on problem solving. Journal of Experimental Psychology, 63, 12-18.
- Goel, V. & Grafman, J. (1995) Are the frontal lobes implicated in "planning" functions? Interpreting data from the Tower of Hanoi. Neuropsychologica, 33, 623-642.
- Hayes, J. R. & Simon, H. A. (1974). Understanding written problem instructions. In L. W. Gregg (Ed.) Knowledge and cognition. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Hayes, J. R. & Simon, H. A. (1977). Psychological differences among problem isomorphs. In N. J. Castellan, D. B., Pisoni, G. R. Potts (Eds.), Cognitive theory. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Hinz, A. M. (1992). Shortest paths between regular states of the Tower of Hanoi. Information Sciences, 63, 173-181.
- John, B. E. (1996). Task matters. In D. M. Steier & T. M. Mitchell (Eds.), Mind matters: A tribute to Allen Newell, 313-324. Hillsdale, NJ: Erlbaum.

Klahr, D. & Robinson, M. (1981). Formal assessment of problem solving and planning processes on preschool children. Cognitive Psychology, 13, 113-148.

Kotovsky, K. & Fallside, D. (1989). Representation and transfer in problem solving. In D. Klahr & K. Kotovsky (Eds.), Complex information processing: The impact of Herbert A. Simon. The 21st Carnegie-Mellon Symposium on Cognition, 69-107. Hillsdale, NJ: Lawrence Erlbaum Associates.

Kotovsky, K., Hayes, J. R., & Simon, H. A. (1985). Why are some problems hard? Evidence from Tower of Hanoi. Cognitive Psychology, 17, 248-294.

Newell, A. & Simon, H. A. (1972). Human problem solving. Englewood Cliffs, NJ: Prentice-Hall.

Ruiz, D. (1987). Learning and problem solving: What is learned while solving the Tower of Hanoi. (Doctoral Dissertation, Stanford University). Dissertation Abstracts International, 42, 3438B.

Shallice, T. (1982) Specific impairments of planning. Phil. Trans. Royal Society of London, 199, 199-209.

Simon, H. A. & Hayes, J. R. (1976), The understanding process: Problem isomorphs. Cognitive Psychology, 8, 165-190.

Simon, H. A. (1975). The functional equivalence of problem solving skills. Cognitive Psychology, 7, 268-288.

Suchman, L. A. (1987). Plans and situated action: The problem of human-machine communication. New York: Cambridge University Press.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem solving strategies. Cognitive Science, 15(1), 1-48.

Ward, G. & Allport, A. (1997). Planning and problem solving using the five-disc Tower of London task. The Quarterly Journal of Experimental Psychology, 50A, 49-78.

Young, R. M. & Lewis, R. L. (1999). The Soar cognitive architecture and human working memory. In A. Miyake & P. Shah (Eds.), Models of working memory: Mechanisms of active maintenance and executive control, pp. 224-256. New York: Cambridge University Press.

Footnotes

1 - This research is supported by grant BCS997-5-220 from the National Science Foundation. We would like to thank Christian Lebiere and Alex Petrov for their comments on this manuscript.

2 - See Figure 3 for examples of non tower-to-tower problems. The goal recursion strategy will not apply to these because it only involves setting subgoals to achieve tower configurations.

3 - There are 3 positions where the 4-disk can start, 2 different places for it to go in the goal configuration, 2 ways for the 3-disk to block that move, 2 ways for the 2-disk to block moving the 3-disk out of the way and 2 ways for the 1-disk to block moving the 2-disk out of the way. After the 4-disk has been moved to its destination, there will be a pyramid of the 3 smaller disks and 2 places in the goal configuration for the 3-disk to go that will be blocked by the 1-disk and the 2-disk. After the 3-disk has been moved to its destination there will be a pyramid of the 2 smallest disks and 2 places for the 2-disk to go that will be blocked by the 1-disk. After the 2-disk has been moved to its destination there will be 2 destinations for the 1-disk that will require moving it. Thus, there are 3×2^7 possible problems with the same subgoal structure.

4 - The average time over participants was calculated for each of the 10 actions in each of the 6 End sequences. The correlation across the 10 actions was calculated for each of the 15 pairs of End sequences. This number reported is the average of these 15 correlations.

Table 1
Application of the Sophisticated Perceptual Strategy
to the Problems in Figure 1

(a) Cognitive Steps in solving problem. Goal posts and disk moves are numbered to facilitate making the correspondence with parts (b) and (c) of this table.

	Action Number	Action
Encode Problem		
Formulate goal of moving 4 to C and store it	P1	post 4
Formulate goal of moving 3 to B and store it	P2	post 3
Formulate goal of moving 2 to C and store it	P3	post 2
Formulate goal of moving 1 to B and do it	M1	move 1
Retrieve goal of moving 2 to C and do it	M2	move 2
Retrieve goal of moving 3 to B		
Formulate goal of moving 1 to C and do it	M3	move 1
Retrieve goal of moving 3 to B and do it	M4	move 3
Retrieve goal of moving 4 to C		
Formulate goal of moving 2 to B and store it	P4	post 2
Formulate goal of moving 1 to A and do it	M5	move 1
Retrieve goal of moving 2 to B and do it	M6	move 2
Retrieve goal of moving 4 to C		
Formulate goal of moving 1 to B and do it	M7	move 1
Retrieve goal of moving 4 to C and do it	M8	move 4
Formulate goal of moving 3 to C and store it	P5	post 3
Formulate goal of moving 2 to A and store it	P6	post 2
Formulate goal of moving 1 to C and do it	M9	move 1
Retrieve goal of moving 2 to A and do it	M10	move 2
Retrieve goal of moving 3 to C		
Formulate goal of moving 1 to A and do it	M11	move 1
Retrieve goal of moving 3 to C and do it	M12	move 3
Formulate goal of moving 2 to C and store it	P7	post 2
Formulate goal of moving 1 to B and do it	M13	move 1
Retrieve goal of moving 2 to C and do it	M14	move 2
Formulate goal of moving 1 to C and do it	M15	move 1

(b) Cognitive operations associated with each action in the traditional Tower of Hanoi problem.
The numbers appearing in the store column refer to the number of subgoals stored.

Action Number	Action	Encode	Formulate	Store	Retrieve	Time (sec.)
M1	Move 1	1	4	3	0	9.7
M2	Move 2	0	0	0	1	2.5
M3	Move 1	0	1	0	1	2.8
M4	Move 3	0	0	0	1	2.2
M5	Move 1	0	2	1	1	3.1
M6	Move 2	0	0	0	1	2.0
M7	Move 1	0	1	0	1	2.4
M8	Move 4	0	0	0	1	2.0
M9	Move 1	0	3	2	0	6.6
M10	Move 2	0	0	0	1	2.4
M11	Move 1	0	1	0	1	2.9
M12	Move 3	0	0	0	1	2.1
M13	Move 1	0	2	1	0	5.4
M14	Move 2	0	0	0	1	2.2
M15	Move 1	0	1	0	0	2.0
Correlations						
	Formulate	0.66				
	Store	0.77	0.93			
	Retrieve	-0.44	-0.75	-0.70		
	Time	0.80	0.92	0.97	-0.73	
Correlations without Action 1						
	Store		0.89			
	Retrieve		-0.67	-0.63		
	Time		0.86	0.92	-0.69	

(c) Cognitive operations associated with each action in the augmented Tower of Hanoi problem. The numbers appearing in the store and retrieve columns refer to the number of intervening actions.

Action Number	Action	Encode	Formulate	Store	Retrieve	Time
P1	Post 4	1	1	6	0	5.11
P2	Post 3	0	1	3	0	2.77
P3	Post 2	0	1	1	0	2.63
M1	Move 1	0	1	0	0	2.77
M2	Move 2	0	0	0	1	2.67
M3	Move 1	0	1	0	3	2.91
M4	Move 3	0	0	0	0	2.90
P4	Post 2	0	1	1	6	3.12
M5	Move 1	0	1	0	0	2.65
M6	Move 2	0	0	0	1	2.77
M7	Move 1	0	1	0	2	2.75
M8	Move 4	0	0	0	0	2.58
P5	Post 3	0	1	3	0	2.94
P6	Post 2	0	1	1	0	2.73
M9	Move 1	0	1	0	0	2.69
M10	Move 2	0	0	0	1	2.64
M11	Move 1	0	1	0	3	2.89
M12	Move 3	0	0	0	0	2.55
P7	Post 2	0	1	1	0	2.80
M13	Move 1	0	1	0	0	2.65
M14	Move 2	0	0	0	1	2.65
M15	Move 1	0	1	0	0	2.64
Correlations						
	Formulate	0.15				
	Store	0.79	0.34			
	Retrieve	-0.12	0.11	-0.15		
	Time	0.97	0.24	0.82	0.06	
Correlations without Action 1						
	Store		0.37			
	Retrieve		0.14	-0.09		
	Time		0.36	0.36	0.68	

Table 2
Actions Involving the 4 and 5 Disks
Classified according to whether they involved posting or moving,
the size of the desk, and whether it was the initial action or internal
to the action sequence.

	Latency	Accuracy
Post 5 (Initial)	4.72 sec	88.5%
Move 5 (Internal)	2.66 sec	95.3%
Post 4 (Initial)	5.02 sec	81.5%
Post 4 (Internal)	3.02 sec	90.6%
Move 4 (Internal)	2.77 sec	91.1%

Table 3
Cognitive operations associated with the 30 critical actions

Category	Action	Formulate	Store	Retrieve	Time
Under 4	Post 3	1	3	0	2.85
	Post 2	1	1	0	2.69
	Move 1	1	0	0	2.72
	Move 2	0	0	1	2.69
	Move 1	1	0	3	3.02
	Move 3	0	0	0	2.88
	Post 2	1	1	6	3.43
	Move 1	1	0	0	2.73
	Move 2	0	0	1	2.66
	Move 1	1	0	2	2.84
Under 5	Post 3	1	3	12	3.82
	Post 2	1	1	0	2.72
	Move 1	1	0	0	2.71
	Move 2	0	0	1	2.73
	Move 1	1	0	3	3.09
	Move 3	0	0	0	2.86
	Post 2	1	1	6	3.35
	Move 1	1	0	0	2.73
	Move 2	0	0	1	2.67
	Move 1	1	0	2	2.83
End	Post 3	1	3	0	2.96
	Post 2	1	1	0	2.74
	Move 1	1	0	0	2.71
	Move 2	0	0	1	2.64
	Move 1	1	0	3	2.91
	Move 3	0	0	0	2.65
	Post 2	1	1	0	2.81
	Move 1	1	0	0	2.69
	Move 2	0	0	1	2.64
	Move 1	1	0	0	2.71
	Correlations				
	Store	0.35			
	Retrieve	0.20	0.51		
	Time	0.37	0.67	0.93	

Figure Captions

Figure 1. The four-disk version of the Tower of Hanoi problem.

Figure 2: An example of the problem screen that participants saw. The grid was not seen by participants but reflects separation of approximately 1 degree of visual angle.

Figure 3: Examples of 4-disk problems that are 15 moves apart.

Figure 4: Mean and predicted latencies for the three types of 10-action sequences: (a) Under 4; (b) Under 5; (c) End.

Figure 5: Mean and predicted proportion of perfect moves for the three types of 10-action sequences: (a) Under 4; (b) Under 5; (c) End.

Figure 6: Boxes indicate the regions of interest (ROIs) used in analyzing the screen.

Figure 7: Mean proportion of time spent fixating the planning regions for the three types of 10-action sequences.

Figure 1

Figure 9.9 from L&M

Figure 2

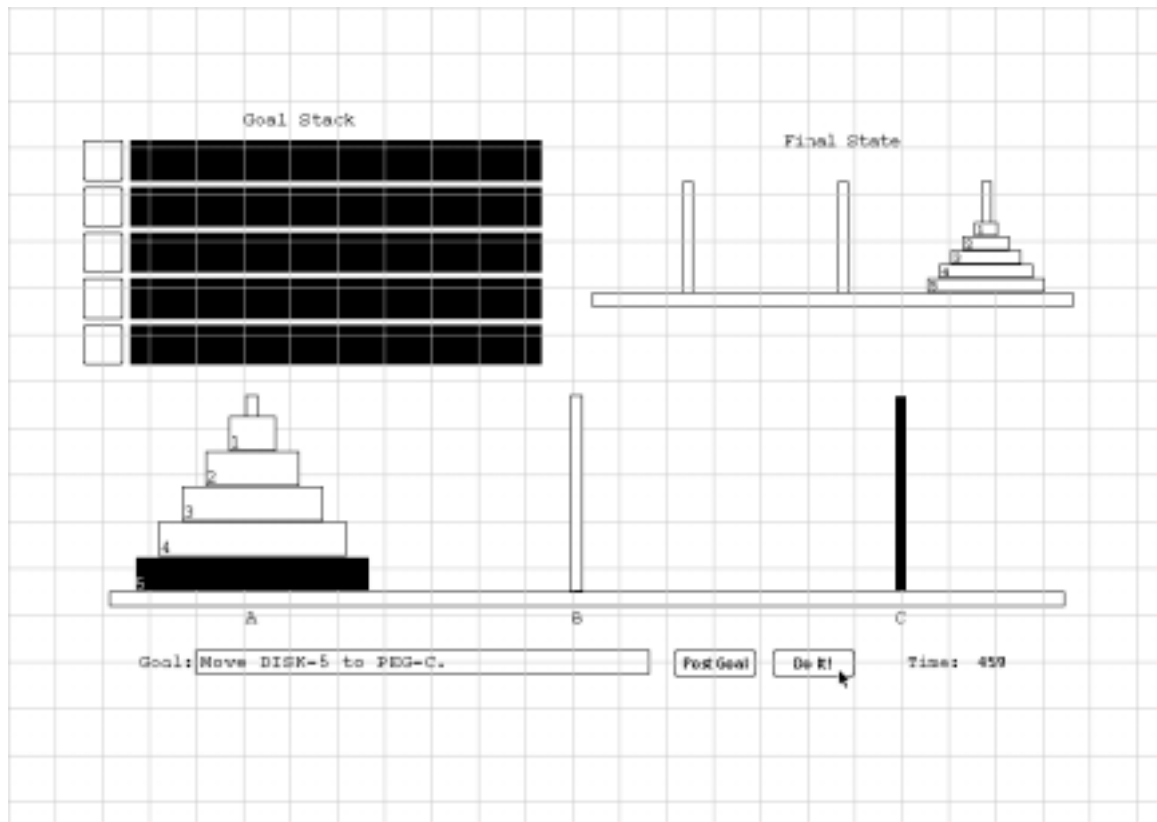
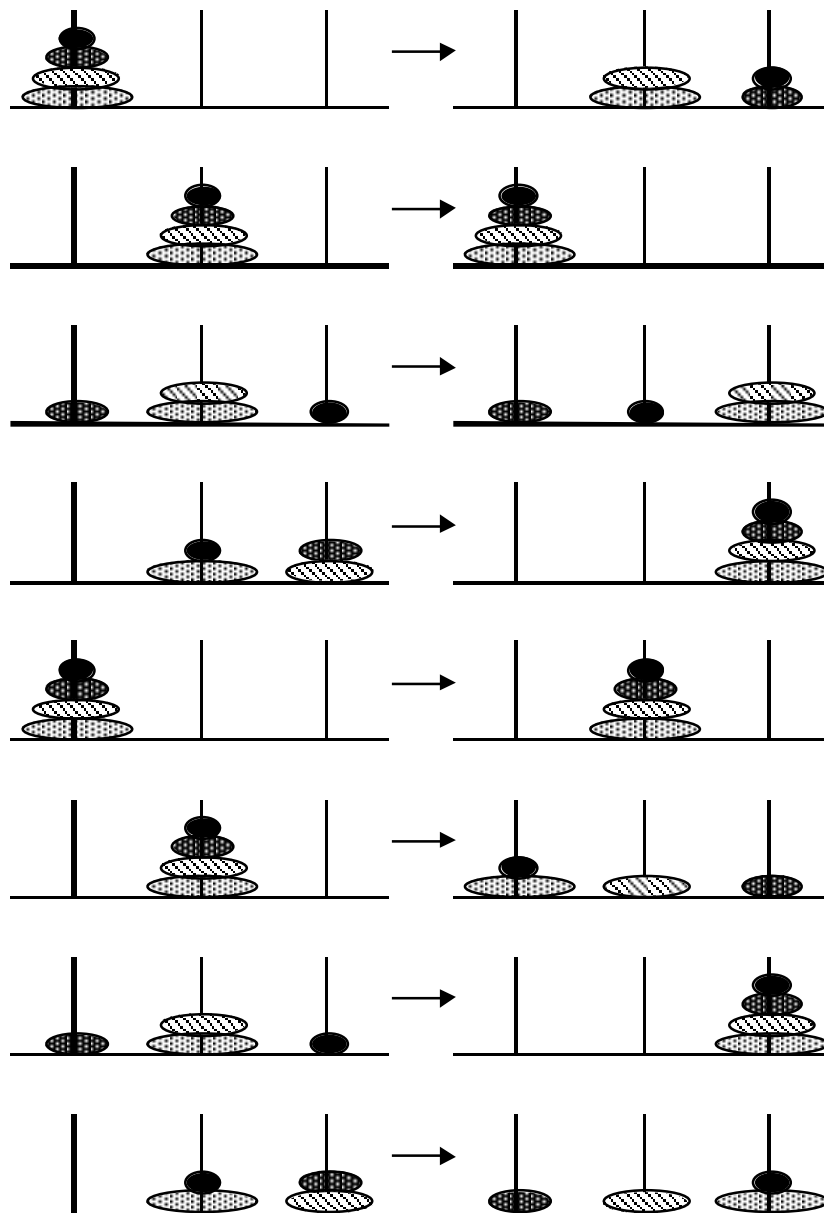
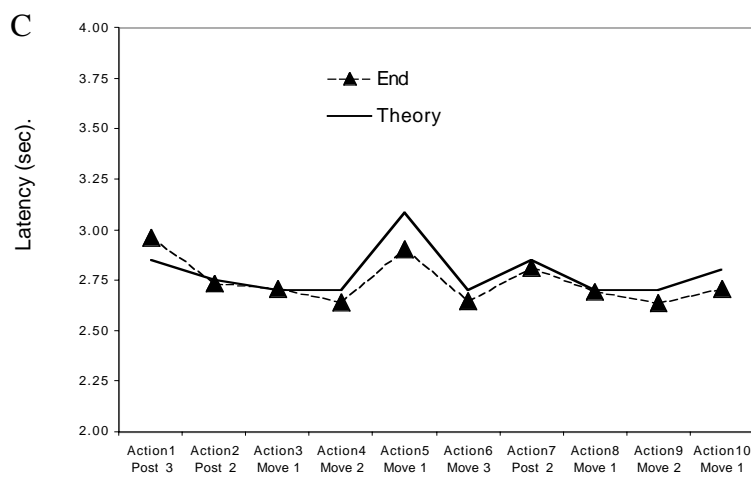
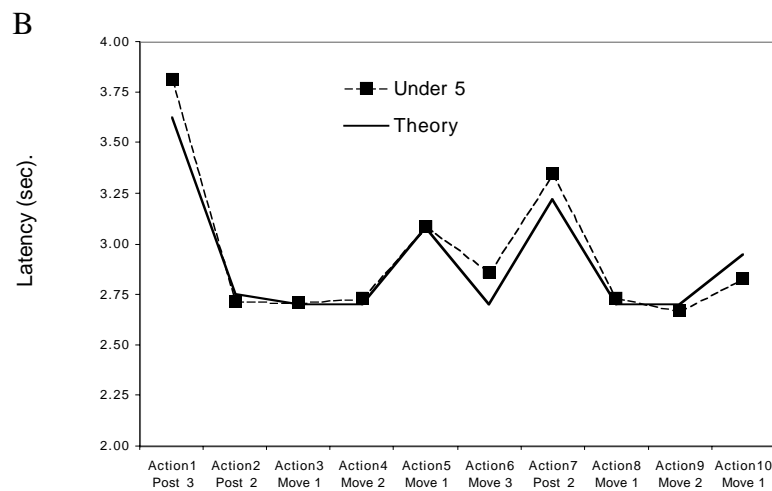
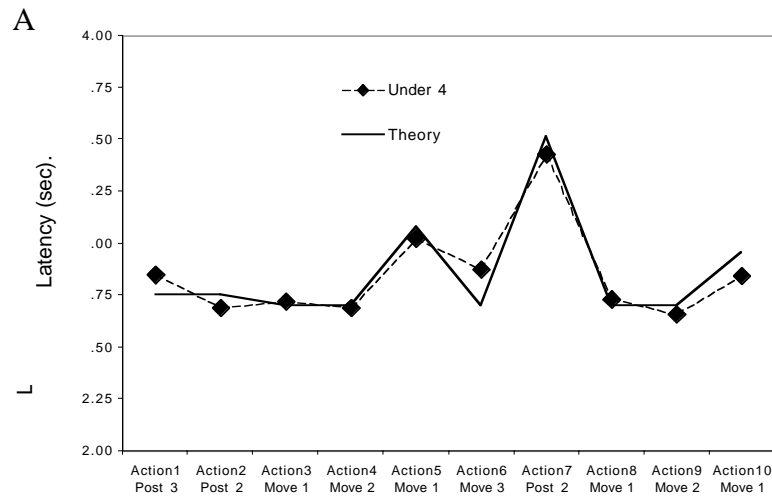


Figure 3





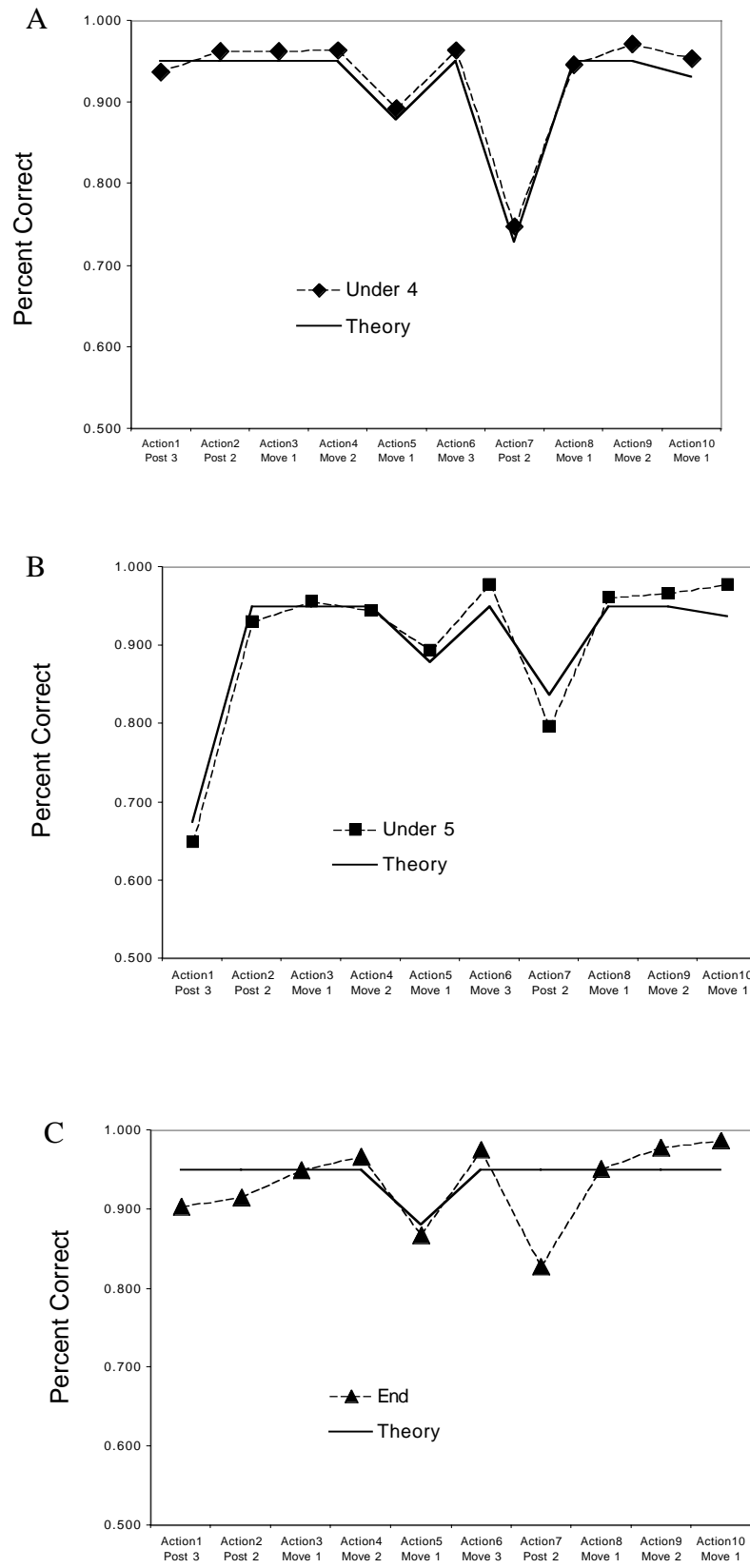


Figure 6

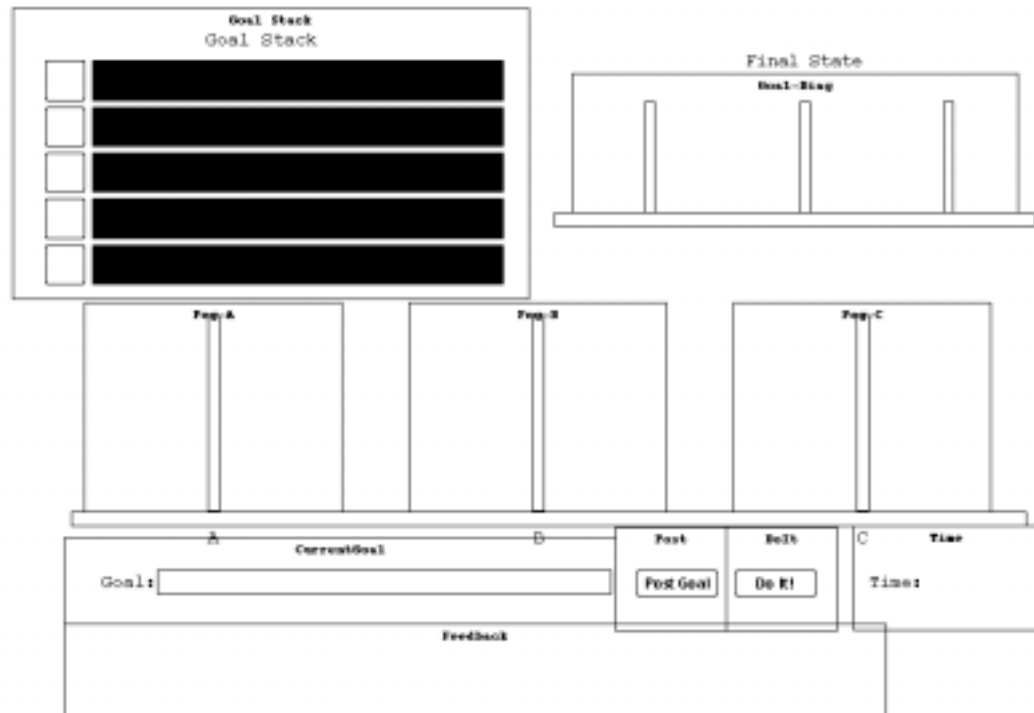
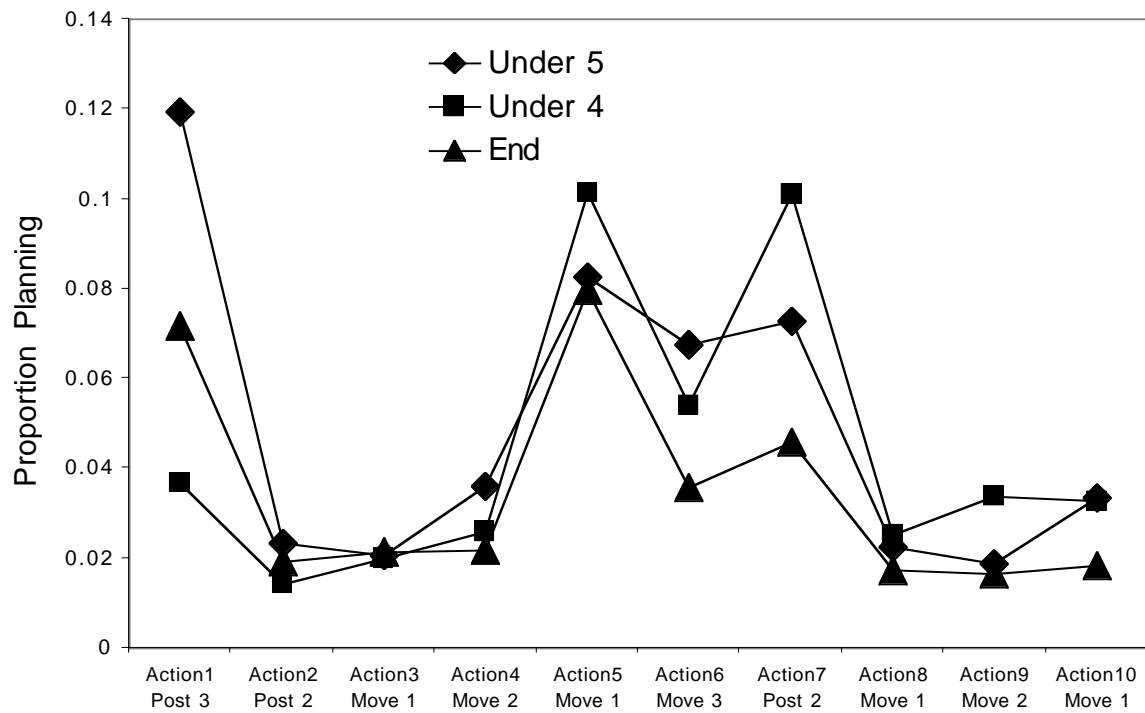


Figure 7



¹ This research is supported by grant BCS997-5-220 from the National Science Foundation. We would like to thank Christian Lebiere and Alex Petrov for their comments on this manuscript.

² See Figure 2 for an example of non tower-to-tower problems.

³ There are 3 positions where the 4-disk can start, 2 different places for it to go in the goal configuration, 2 ways for the 3-disk to block that move, 2 ways for the 2-disk to block moving the 3-disk out of the way and 2 ways for the 1-disk to block moving the 2-disk out of the way. After the 4-disk has been moved to its destination, there will be a pyramid of the 3 smaller disks and 2 places in the goal configuration for the 3-disk to go that will be blocked by the 1-disk and the 2-disk. After the 3-disk has been moved to its destination there will be a pyramid of the 2 smallest disks and 2 places for the 2-disk to go that will be blocked by the 1-disk. After the 2-disk has been moved to its destination there will be 2 destinations for the 1-disk that will require moving it. Thus, there are 3×2^7 possible problems with the same subgoal structure.

⁴ The average time over participants was calculated for each of the 10 actions in each of the 6 **End** sequences. The correlation across the 10 actions was calculated for each of the 15 pairs of **End** sequences. This number reported is the average of these 15 correlations.