A COMPUTATIONAL MODEL OF TOWER OF HANOI PROBLEM SOLVING

By

Sashank Varma

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Psychology

May, 2006

Nashville, Tennessee

Approved:

Professor Timothy P. McNamara

Professor Susan R. Goldman

Professor John D. Bransford

Professor Gordon D. Logan

Professor David C. Noelle

To my father, Sreekanth

TABLE OF CONTENTS

Appendix

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The Tower of Hanoi (TOH) is a classic task in cognitive science. During the 1960s and 1970s, it served as a testbed for theories of problem solving, knowledge representation, skill acquisition, and transfer (Anzai & Simon, 1979; Kotovsky, Hayes, & Simon, 1985; Simon, 1975; Simon & Hayes, 1976). Over the past twenty-five years, the TOH task has found broader applications. It has been used to document the development and decline of cognition over the entire human lifespan (Brennan, Welsh, & Fisher, 1997; Klahr & Robinson, 1981). It has proven to be a diagnostic neuropsychological test of the cognitive deficits of patients (e.g., Glosser & Goodglass, 1990). More recently, it has been used in functional neuroimaging studies of the neural basis of problem solving (Fincham, Carter, van Veen, Stenger, & Anderson, 2002). Through these applications, the TOH task has been recast as a lens on strategic, high-level thinking. In this new role, the TOH task has returned to the forefront of cognitive science.

As the TOH task has spread from young adults to children and the elderly, from intact normals to patients with frontal lobe lesions, and from behavioral to neuroimaging investigations, theoretical and computational accounts of the underlying mental processes have failed to keep pace. A comprehensive account of task performance is currently lacking, and the likelihood that one will emerge decreases as empirical investigations grow more disconnected.

The primary achievement of this dissertation has been to fill this gap. A new computational model has been constructed that accounts for a broad range of the new data on TOH problem solving. It consists of three levels. At the bottom is an architectural foundation. At the top are those representations and processes specific to the TOH task itself. In the middle is an account of the interaction between frontal and parietal areas that bridges between the other levels in a principled way. The model has been evaluated against behavioral data collected from normal young adults, behavioral data collected from patients with frontal lobe lesions, and neuroimaging data collected from normal young adults.

There have been a number of secondary achievements as well. One is a new account of how goals are organized and how they control cognition. For the first forty years of the cognitive revolution, it was widely assumed that goals are organized as a stack of arbitrary depth, with only the topmost (i.e., most recent) goal guiding problem solving. This assumption has been challenged in recent years (Altmann & Trafton, 2002; Anderson & Douglass, 2001; Just, Carpenter, & Hemphill, 1996). The model described below represents the finest deconstruction of the goal stack yet proposed, one that makes strong claims about neural localization.

Another achievement is a novel account of selection, one that reconciles two computational mechanisms that appear at first glance to be incommensurable: the preference machinery of the Soar cognitive architecture (Newell, 1990) and the contention scheduler of Shallice's (1982) theory of executive function. The result has no need for a homunculus.

Another achievement is an account of fronto-parietal interaction that synthesizes existing theories of problem solving and executive function. According to the account, the executive functions of problem solving – the management of goals and the selection between alternatives – are localized in frontal areas such as dorsolateral prefrontal cortex (DLPFC). Executive function is a topic of much research in cognitive psychology, neuropsychology, and neuroscience. Most executive function tasks, such as the Stroop task, are low-level: light in their computational demands and short in duration. Problem solving, by contrast, is a high-level consumer of executive function, one that makes heavy computational demands over the course of tens of seconds (or longer). The non-executive components of problem solving are domain-specific. In particular, the TOH task is visuospatial in nature (versus, say, logical deduction, which is comparatively linguistic). The visuospatial aspects of problem solving are localized in parietal areas such as intra-parietal sulcus (IPS). The account of fronto-parietal interaction offered here explains how left and right DLPFC and left and right IPS collaborate in the service of high-level cognition.

Another achievement is methodological. Every theory or model faces the problem of degrees of freedom. Degrees of freedom typically take the form of numerically-valued free parameters. Although the model contains such parameters, they are regarded as scientifically peripheral. Instead, it is the *design decisions* made during the construction process that are the model's most interesting degrees of freedom. There are five such design decisions, four binary-valued and one ternary-valued. They define a space of 48 model variants that are evaluated against the data. The result of interest is not best-fitting values for the model's free parameters. Rather, it is guidance on four of the model's five design decisions, or equivalently, a reduction in the space of model variants that need to be considered in future research.

A final achievement is further validation of the 4CAPS cognitive architecture (Just & Varma, 2006). Like all cognitive architectures, 4CAPS purports to be a unified account of all domains of (high-level) cognition. It has supported successful models of sentence comprehension, mental rotation, driving, and complex dual-tasking, among other tasks. This dissertation project extends its scope to the domain of TOH problem solving.

The remainder of this dissertation has the following structure. Chapters II, III, and IV serve as background: Chapter II introduces the TOH task and describes several strategies for solving

TOH problems; Chapter III summarizes the results of recent empirical investigations of TOH problem solving; and Chapter IV reviews recent computational models of TOH problem solving. Chapters V, VI, and VII describe the three levels of the TOH model. Chapter V describes the bottom level, the 4CAPS cognitive architecture. Chapter VI describes the middle level, the model of fronto-parietal interaction that synthesizes existing theories of problem solving and executive function. Chapter VII describes the top level, where the fronto-parietal model is specialized for the TOH task. Chapters VIII, IX, and X describe the model's correspondence to the data on TOH problem solving: Chapter VIII addresses behavioral measures collected from normal young adults; Chapter IX behavioral measures collected from patients with frontal lobe lesions; and Chapter X neuroimaging measures collected from normal young adults. Finally, Chapter XI summarizes the achievements of this dissertation project and Chapter XII sketches three paths for future exploration.

CHAPTER II

THE TOH TASK

Simon has called the TOH task the *drosophila* of cognition. Although this is an overstatement, it is certainly a signature task of problem solving, and has found wide application in domains such as working memory, intelligence, executive function, and frontal lobe function. This chapter introduces the TOH task and attendant terminology. It then describes a number of strategies for solving TOH problems, indicating which have found empirical support in the literature.

Task Definition

A TOH problem involves N disks of graduated size, with disk 1 the smallest and disk N the largest. A problem is defined by a starting configuration and an ending configuration. Each configuration consists of three pegs – left, middle, and right – and a distribution of the N disks across the pegs such that larger disks are never on top of smaller disks. The goal is to transform the starting configuration into the ending configuration via a sequence of moves. A move transfers one disk from a source peg to a destination peg, subject to three restrictions: (1) one and only one disk is moved; (2) the disk is on top of the source peg at the beginning of the move and on top of the destination peg at the end of the move; and (3) the disk is not moved on top of a smaller disk. A move that that satisfies all three criteria is deemed legal. A solution sequence for a problem is a sequence of legal moves that transforms the starting configuration into the ending configuration. An optimal solution sequence is one of minimal length. The optimal solution sequence for an N-disk problem is no longer than 2^N-1 moves.

An N-disk TOH problem can have any one of 3^N starting configurations and 3^N-1 ending configurations, and therefore there are a total of $3^N(3^N-1)$ problems. It is useful to group TOH problems into five classes based on the spatial properties of their starting and ending configurations. Figure 1 shows one problem from each class for the case where N=3. In the standard tower-to-tower problem, all N disks are stacked on the left peg in the starting configuration and all N disks stacked on the right peg in the ending configuration. There is only one standard tower-to-tower problem for each value of N. The second class of problems is slightly larger: tower-to-tower problems have all N disks stacked on one peg in the starting configuration and all N disks are stacked on another peg in the ending configuration. For a given N, this class contains six problems, one of which is the standard tower-to-tower problem. All tower-to-tower problems (including the standard one) require the maximum 2^N-1 moves to solve. Tower-to-spread problems have all N disks stacked on one peg in the starting configuration; in the ending configuration, the disks are distributed over two or more pegs. Spread-to-tower problems are just the opposite: the disks are spread over two or more pegs in the starting configuration and stacked on one peg in the ending configuration. Finally, spread-to-spread problems distribute the disks over two or more pegs in both the starting and ending configurations. A variable number of moves – between 1 and 2^N-1 – are required to solve problems of the last three classes. (The exact number depends on the specific starting and ending configurations.)

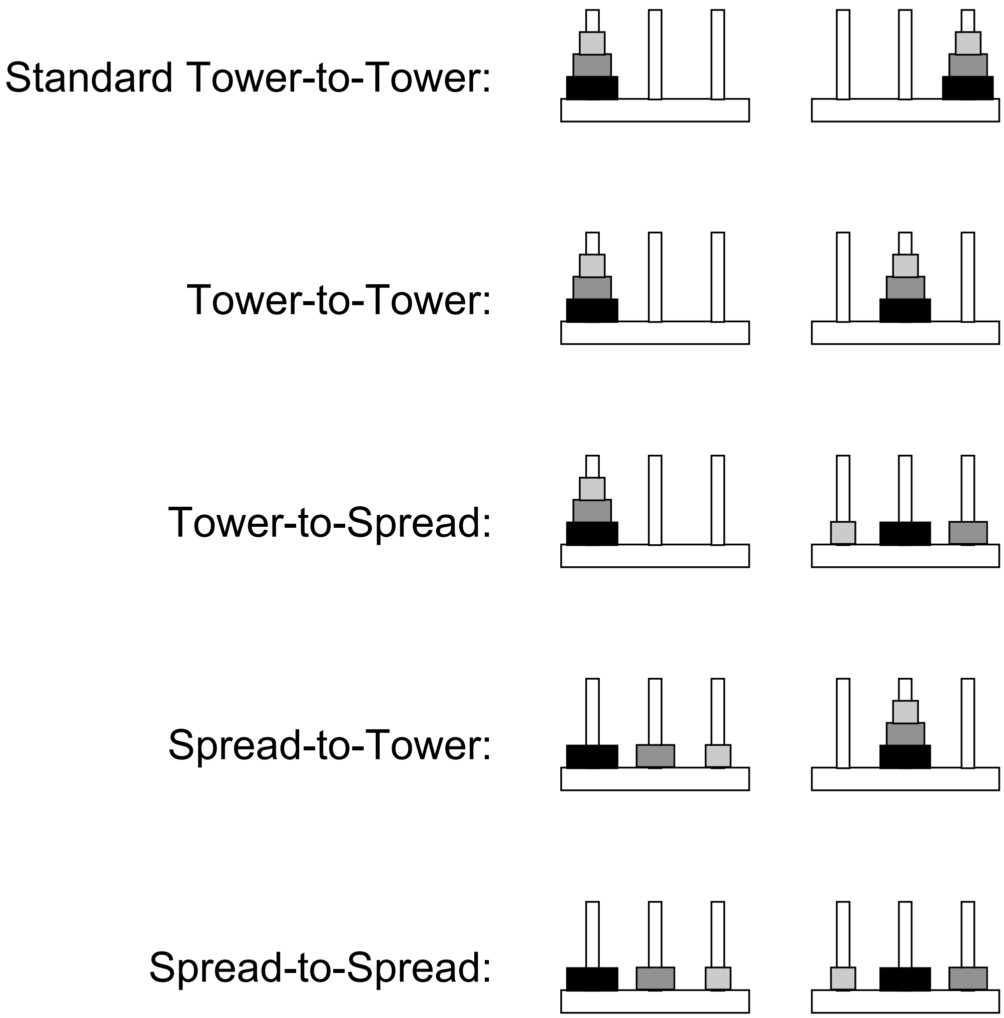


Figure 1. The five classes of TOH problem.

Solution Strategies

Simon (1975) carried out a detailed analysis of the TOH task, identifying a number of logically possible solution strategies. This analysis has framed subsequent empirical research, and it is therefore worth reviewing the strategies in some detail. To simplify matters, each will be described in the context of the standard *N*-disk tower-to-tower problem. The model developed in subsequent chapters implements the second of these strategies, the sophisticated perceptual strategy.

Goal Recursion

The *goal recursion strategy* is the most abstract of Simon's (1975) strategies in that it employs two concepts that are not part of the task instructions themselves. The first is that notion of a stack of *n* disks (an *n*-stack), which generalizes the notion of a single disk. The second is the notion of moving a stack of *n* disks, which generalizes the notion of moving a single disk. The second of these concepts can be implemented by the following recursive algorithm:

To move a stack of *n* disks from peg *A* to peg *B*:
(1) If $n>1$, then move the stack of the n-1 smallest disks from peg *A* to peg *C*.
(2) Move disk *n* (the largest remaining one) from peg *A* to peg *B*.
(3) If $n>1$, then move the stack of the n-1 smallest disks from peg *C* to peg *B*.

Step (2) is well-defined because moving a single disk from one peg to another is licensed by the task instructions themselves. The problem is how to perform steps (1) and (3) given the prohibition against moving more than one disk at a time. The solution is to recursively employ the same algorithm to perform these steps, but for the smaller stack of n-1 disks (and of course for different pegs). Recursive decomposition continues until $n=1$ (i.e., the stack consists of a single disk) at which point the move can be directly performed without the need for recursion.

With the abstract notions of "stack" and "moving a stack" in hand, solving the standard *N*-disk tower-to-tower problem reduces to moving a stack of *N* disks from the left peg to the right peg.

Executing the goal recursion strategy requires keeping track of one's place in the recursive decomposition of larger stacks into smaller stacks. This is done by using a stack of goals. To see how this works, consider the trace of the goal recursion strategy on the standard 3-disk tower-to-tower problem.

Task Goal: Move the 3-stack from the left peg to the right peg.
(1) Goal: Move the 2-stack from the left peg to the middle peg.
    (1) Goal: Move the 1-stack from the left peg to the right peg.
        (2) Move disk 1 from the left peg to the right peg.
    (2) Move disk 2 from the left peg to the middle peg.
    (3) Goal: Move the 1-stack from the right peg to the middle peg.
        (2) Move disk 1 from the right peg to the middle peg.
(2) Move disk 3 from the left peg to the right peg.
(3) Goal: Move the 2-stack from the middle peg to the right peg.
    (1) Goal: Move the 1-stack from the middle peg to the left peg.
        (2) Move disk 1 from the middle peg to the left peg.
    (2) Move disk 2 from the middle peg to the right peg.
    (3) Goal: Move the 1-stack from the left peg to the right peg.
        (2) Move disk 1 from the left peg to the right peg.

The trace illustrates the three distinctive properties of the goal recursion strategy. The first is its use of abstractions – stacks of disks and their movement – not defined in the task instructions themselves. Problem solvers do not know these abstractions when they first approach TOH problems, and therefore do not spontaneously apply the goal recursion strategy. It must be induced through extensive practice, typically in conjunction with explicit instruction. Second, the goal recursion strategy makes extensive use of goals. Not all solution strategies use goals, and of those that do, none use them as heavily. Third, the only configurations the goal recursion strategy considers are the starting and ending configurations that define the problem being solved. In particular, the intermediate configurations generated during the course of problem solving are not consulted when determining the next move. In this regard, the goal recursion strategy differs from the perceptual strategies described next.

The Sophisticated Perceptual Strategy

The goal recursion strategy, like Tolman's rat, is lost in its own thoughts. It requires little visuospatial contact with the external environment. This is the key difference between it and the perceptual strategies. The sophisticated perceptual strategy focuses on the largest out-of-place disk in the current configuration. It attempts to move this disk to its peg position in the ending configuration. If the source peg is blocked (i.e., the largest out-of-place disk is covered by another disk) or the destination peg is blocked (i.e., contains another disk), then a goal is established to clear the largest blocking disk by moving it to the remaining peg. (This non-source, non-destination peg is called the buffer peg.) Clearing a blocking disk can require clearing additional blocking disks, a recursive process. To manage this recursion, the sophisticated perceptual strategy also uses goals, although less heavily than the goal recursion strategy. When the largest out-of-place disk has been unblocked, it is moved. This process is then repeated for the largest remaining out-of-place disk. If no such disk exists (i.e., the current and ending configurations are identical), then the problem has been solved and the algorithm halts. More formally:

To transform the current configuration into the ending configuration:

(1) Compare the current and ending configurations. If they are the same, then the problem has been solved. Otherwise, identify the largest out-of-place disk *n*.

(2) Identify the location of disk *n* in the current configuration (peg *A*). Identify the location of *n* in the ending configuration (peg *B*). Identify the largest disk *k* in the current configuration blocking *n*'s movement. (Disk *k* will be on top of disk *n* or will occupy its intended position on peg *B*.)

(3) If a blocking disk *k* exists, then move it from its current peg (*A* or *B*) to peg *C*. To do this, go to step (2), but with *n* bound to *k* and the destination location *B* bound to *C*.

(4) If no blocking disk exists in the current configuration, then move disk *n* from peg *A* to peg *B*, producing a new current configuration. Go to step (1).

The sophisticated perceptual strategy is applied to a problem by initializing the current configuration to the problem's starting configuration.

As noted above, the sophisticated perceptual strategy requires a goal stack to organize the recursive clearing of blocking disks, but one that is shallower on average than the one required by the goal recursion strategy. This difference is illustrated by tracing the sophisticated perceptual strategy on the standard 3-disk tower-to-tower problem.

Task Goal: Solve the problem where the current (i.e., starting) configuration has a 3-stack on the left peg and the ending configuration a 3-stack on the right peg.
(1) Identify disk 3 as the largest out-of-place disk in the current configuration.
(2) Identify disk 2 as the largest blocking disk in the current configuration.
(3) Goal: Move disk 2 from the left peg to the middle peg.
    (2) Identify disk 1 as the largest blocking disk in the current configuration.
    (3) Goal: Move disk 1 from the left peg to the right peg.
        (4) Move disk 1 from the left peg to the right peg.
    (4) Move disk 2 from the left peg to the middle peg.
(1) Identify disk 3 as the largest out-of-place disk in the current configuration.
(2) Identify disk 1 as the largest blocking disk in the current configuration.
(3) Goal: Move disk 1 from the right peg to the middle peg.
    (4) Move disk 1 from the right peg to the middle peg.
(1) Identify disk 3 as the largest out-of-place disk in the current configuration.
(4) Move disk 3 from the left peg to the right peg.
(1) Identify disk 2 as the largest out-of-place disk in the current configuration.
(2) Identify disk 1 as the largest blocking disk in the current configuration.
(3) Goal: Move disk 1 from the middle peg to the left peg.
    (4) Move disk 1 from the middle peg to the left peg.
(1) Identify disk 2 as the largest out-of-place disk in the current configuration.
(4) Move disk 2 from the middle peg to the right peg.
(1) Identify disk 1 as the largest out-of-place disk in the current configuration.
(4) Move disk 1 from the left peg to the right peg.

Comparing the traces of the sophisticated perceptual strategy and the goal recursion strategy reveals four important differences. First, the sophisticated perceptual strategy interacts frequently with the external environment. Specifically, the current configuration is consulted to identify the largest out-of-place disk in step (1) and to identify the largest blocking disk in step (2). By contrast, the goal recursion strategy never consults the current configuration. Second, the sophisticated perceptual strategy establishes fewer goals than the goal recursion strategy, and they stack less deeply. This savings is possible because it evenly balances goal-driven processing and perceptually-driven processing: the strategy is "sophisticated" in maintaining a goal stack to structure the clearing of blocking disks, and it is "perceptual" in searching the current configuration for out-of-place and blocking disks. Third, the sophisticated perceptual strategy is capable of solving TOH problems of all five classes, whereas the goal recursion strategy only applies to tower-to-tower problems. Finally, the sophisticated perceptual strategy is defined in the vocabulary of the TOH task instructions. It does not depend on the abstract notions (e.g., stacks of disks) required by the goal recursion strategy. It is therefore not surprising that novice problem solvers often spontaneously induce this strategy (or its simpler sibling, described next) after solving a small number of practice problems.

The Simple Perceptual Strategy

The simple perceptual strategy is a lobotomized variant of the sophisticated perceptual strategy. Both strategies focus on the largest out-of-place disk in the current configuration, attempting to move it to its peg position in the ending configuration. They differ, however, in how they handle blocking disks. The sophisticated perceptual strategy clears them systematically through the use of a goal stack. The simple perceptual strategy, by contrast, makes no use of goals, relying solely on perceptual heuristics. More formally:

To transform the current configuration into the ending configuration:

(1) Compare the current and ending configurations. If they are the same, then the problem has been solved. Otherwise, randomly select an out-of-place disk *n*.

(2) Identify the location of disk *n* in the current configuration (peg *A*). Identify the disks k_1, k_2, … in the current configuration blocking *n*'s movement. If any exist, randomly select

disk k_i to move; randomly select one of the remaining two pegs to move it to (peg *B* or *C*); perform the move; and repeat step (2).

(3) Identify the location of disk *n* in the ending configuration (peg *B*). Identify the disks k_1, k_2, … in the current configuration blocking this location. If any exist, randomly select disk k_i to move, randomly select one of the remaining two pegs to move it to (peg *A* or *C*); perform the move; and repeat step (3).

(4) Move disk *n* from peg *A* to peg *B*, producing a new current configuration. Go to step (1).

Like its sophisticated sibling, the simple perceptual strategy is applied to a problem by initializing the current configuration to the problem's starting configuration.

The simple perceptual strategy does not use goals to organize the clearing of blocking disks. Instead, when there is a choice between which of multiple blocking disks to move first or which of several pegs to move it to, the choice is made randomly. Chance sometimes smiles on these random choices, as in the following trace of the simple perceptual strategy on the standard 3-disk tower-to-tower problem.

Task Goal: Solve the problem where the current (i.e., starting) configuration has a 3-stack on the left peg and the ending configuration a 3-stack on the right peg.

(1) Randomly select disk 3 as an out-of-place disk in the current configuration.

(2) Identify disks 1 and 2 as on top of disk 3 in the current configuration, blocking its movement. Randomly select disk 1 to move to another peg. Both the middle and right pegs can accommodate it. Randomly choose the right peg, and move disk 1 to it.

(2) Identify disk 2 as on top of disk 3 in the current configuration, blocking its movement. Both the middle and right pegs can accommodate it. Randomly choose the middle peg, and move disk 2 to it.

(3) Identify disk 1 as occupying the destination of disk 3 in the current configuration, blocking its movement. Both the left and middle pegs can accommodate it. Randomly choose the middle peg, and move disk 1 to it.

(4) Move disk 3 from the left peg to the right peg.

(1) Randomly select disk 2 as the largest out-of-place disk in the current configuration.

(2) Identify disk 1 as on top of disk 2 in the current configuration, blocking its movement. Both the left and right pegs can accommodate it. Randomly choose the left peg, and move disk 1 to it.

(4) Move disk 2 from the middle peg to the right peg.

(1) Randomly select disk 1 as the largest out-of-place disk in the current configuration.

(4) Move disk 1 from the left peg to the right peg.

This trace illustrates three properties of the simple perceptual strategy. First, the trace is the same length as the one produced by the goal recursion strategy, and half the length of the trace produced by the sophisticated perceptual strategy. Appearances, in this case, are deceiving. On eight occasions, random choices were made between blocking disks to be moved and between pegs to which to move them. On each occasion, the optimal choice was made. However, when chance is not so favorable (as is typically the case) and suboptimal choices are made, the simple perceptual strategy requires more moves to solve problems than the other strategies. For this reason, it is (much) less likely to produce minimum-length solution sequences. Second, the simple perceptual strategy can be applied to all five classes of TOH problem; in this regard, it is like its sophisticated sibling and unlike the goal recursion strategy. Third, the simple perceptual strategy uses only the terminology of the TOH task instructions. Once again, it is like the sophisticated perceptual strategy in this regard and unlike the goal recursion strategy, which requires the additional, abstract notions of stacks of disks and moving stacks of disks. Not surprisingly then, problem solvers quickly induce the simple perceptual strategy when facing TOH problems for the first time.

Other Strategies

There exist other strategies for solving TOH problems. Only rarely have they proven psychologically relevant, and on those occasions only under highly artificial conditions. They are briefly summarized here.

One such strategy – not mentioned by Simon (1975) – is generate-and-test. This is the simplest artificial intelligence technique for searching through problem spaces – and the least efficient. In the context of the TOH task, it is defined as:

To transform the current configuration into the ending configuration:

(1) Compare the current configuration and ending configuration. If they are the same, then halt; the problem has been solved.

(2) Generate all legal moves from the current configuration.

(3) Randomly select one legal move and apply it to the current configuration, producing a new current configuration. Go to step (1).

Like the perceptual strategies, generate-and-test is applied by initializing the current configuration to the starting configuration of the problem being solved. Because it blindly searches problem spaces, it can require many moves to solve even simple problems. In theory, its poor performance is partially offset by its minimal computational demands – the required perceptual operations are simple, and no goals need to be generated and stored. In practice, however, no experiment has found evidence of its use – not even in extreme populations (i.e., the very young, the very old, and various clinical groups) nor under conditions of severe load.

The move-pattern strategy (Simon, 1975) exploits a topological regularity of the TOH problem space. It is an easy, mechanical strategy to execute because it requires that only one piece of information, the current "parity," be maintained. However, it is highly unlikely that participants can induce this strategy during the course of typical experimental sessions. In fact, no study has found evidence of its spontaneous appearance. The move-pattern strategy can of course be taught, and the earliest (and most grueling) study of TOH problem solving did just that (Ewert & Lambert, 1932). These data are beyond the scope of the proposed dissertation project.

The final strategy mentioned by Simon (1975) is rote memorization. The only study where this strategy might have been used is again Ewert and Lambert (1932). Their participants spent up to three hours solving TOH problems, making thousands of moves! Given the time-on-task and the regularity of the problems (all belonged to the tower-to-tower class), it is possible that some participants memorized solution sequences. However, all subsequent studies have regarded rote memorization as a nuisance strategy and have actively guarded against it by having participants solve many fewer problems that vary in their starting and ending configurations

(Anderson & Douglass, 2001, p. 1336). For this reason, the rote memorization strategy will not be considered further.

CHAPTER III

REVIEW OF THE EMPIRICAL LITERATURE

The TOH task entered psychology in 1932, when Ewert and Lambert first used it to study the effects of verbalization on problem solving. This line of research was taken up again in 1962 by Gagné and Smith (1962), and continues to be pursued even today (Ahlum-Heath & Di Vesta, 1986; Davies, 2000). Over the past three decades, however, most empirical investigations of TOH problem solving have been driven by different research questions.

In the early 1970s, Simon and his colleagues began using the TOH task to investigate the effect of knowledge on problem solving. It proved superior for this purpose than the tasks he and Newell had been using (Newell & Simon, 1972): logical deduction, chess, and cryptarithmetic. These investigations were of two classes. The first explored the relative difficulty of different isomorphs of the TOH task. Isomorphs pose a challenge for Newell and Simon's (1972) theory, which casts problem solving as search through problem spaces, because it predicts that isomorphic tasks – which by definition possess structurally identical problem spaces – should be of equal difficulty. Simon and Hayes (1976) found this prediction to be false, and follow-up work by Kotovsky et al. (1985) and Zhang and Norman (1994) revealed why: Formally isomorphic problem spaces (i.e., states and operators) can differ in their computational demands. These differences cumulate over the course of problem solving, and as result, different TOH isomorphs can vary in difficulty by an order of magnitude.

The second class of experiments focused on the acquisition of new solution strategies. These studies often took a microgenetic approach, sifting the minutiae of individual problem solving sessions for those magical moments when new strategies were spontaneously induced. The pioneering study here is Anzai and Simon (1979), which analyzed in great detail the 90 minute protocol of a single participant who solved the same TOH problem four times. The participant's strategy grew more sophisticated with each attempt, a process Anzai and Simon modeled with an adaptive production system. (Their analysis was later refined by VanLehn (1991).) Gunzelmann and Anderson (2003) recently revisited the transition between solution strategies with experience, documenting this process at both the individual and group levels.

Beginning in the early 1980s, researchers began to lose interest in these research questions, perhaps because they considered them answered. They abandoned experimental paradigms that reveal differences between problem isomorphs and document the induction of new solution strategies, and began aiming new studies at the basic information processing demands of problem solving. These studies typically equate these demands across participants. They employ the standard TOH task, not esoteric isomorphs involving monsters, cups of tea, and orbs of changing size. They also provide explicit instruction on the solution strategy to be employed. In other words, they treat as nuisance variables what were the variables of interest of earlier experiments, focusing instead on the fundamental computational mechanisms of problem solving (e.g., the organization of goals) and their neural implementation (e.g., the effects of a left frontal lesion). It is the results of these contemporary studies that constitute the empirical standard against which the TOH model will be evaluated.

The remainder of this chapter reviews contemporary studies of TOH problem solving. To orient the reader, eight methodological dimensions that organize these studies are first described. The studies themselves are described next, coarsely grouped by the populations they target (normal young adults versus patients with frontal lesions) and the measures they employ (behavioral versus neuroimaging).

Dimensions of Variation

Contemporary studies of TOH problem solving vary on eight methodological dimensions.

The first dimension is the population from which participants are drawn. Most studies use normal young adults with intact brains, i.e., free of lesions and neurodegenerative diseases. A smaller number use patients with lesions to the frontal areas. Only studies that draw from these two populations will be reviewed below. Of course, the TOH task has also been used to study other populations. For example, experiments in the developmental literature have targeted normal children (e.g., Klahr & Robinson, 1981) and elderly participants (e.g., Brennan et al., 1997). The TOH task has also been used to study other clinical populations, including mentally retarded young adults (Spitz, Webster, & Borys, 1982), schizophrenic young adults (e.g., Bustini, Stratta, Daneluzzo, Pollice, Prosperini, & Rossi, 1999), and children diagnosed ADHD (e.g., Aman, Roberts, & Pennington, 1998). The possibility of extending the model to these populations will be discussed in Chapter XII.

The second dimension on which TOH studies vary is the presentation paradigm used. Some constrain problem solving to the optimal solution sequence; others leave participants unconstrained, allowing them to make suboptimal moves. In constrained paradigms, when a suboptimal move is attempted, the experimenter or presentation software flags it as an error, disallows it, and either solicits a replacement move from the participant or makes the optimal move for him or her. The benefit of constrained presentation is that all participants produce the same (optimal) sequence of moves, and therefore their performance can be unambiguously compared on each individual move. The drawback is that constrained presentation lacks ecological validity, straightjacketing natural problem solving behavior. The benefits and drawbacks of constrained presentation are reversed in unconstrained paradigms: although problem solving is more ecologically valid, participants cannot be compared on individual moves because the solution sequences of different participants can differ. A single study has tried for the benefits of both constrained and unconstrained presentation paradigms without their respective drawbacks (Anderson, Kushmerick, & Lebiere, 1993). In this study, participants were unconstrained in their problem solving, lending ecological validity. A large number of problems were employed that possessed isomorphic optimal solution sequences. Therefore, on those occasions when participants solved problems using the optimal move sequence, their performance could be compared on each individual move, as if a constrained presentation paradigm had been used.

The third dimension on which recent studies of TOH problem solving vary is the nature of the problems employed. As described above, TOH problems differ in two ways. The first difference is the number of disks. The problems used by the studies reviewed below used between two and seven disks, and therefore required between 3 and 127 moves to solve optimally. (In more detail, problems with as few as two disks were used as practice problems to familiarize participants with the TOH task and to allow them to induce a solution strategy; the actual data were collected on problems with greater numbers of disks.) The second way in which problems differ is their topological class: standard tower-to-tower, tower-to-tower, tower-to-spread, spread-to-tower, and spread-to-spread. Some studies employed only tower-to-tower problems; others sampled from multiple classes.

The fourth dimension of variation is whether participants are instructed on which solution strategy to use or whether they induce one through the solution of practice problems. Studies that provide explicit instruction control this source of variation between participants, enabling precise calculation of the computational demand of each problem and, when a constrained presentation paradigm is adopted, of each move. The casualty of explicit instruction is ecological validity – there follows a certain artificiality to problem solving. This is problematic if the goal is to investigate the acquisition of new solution strategies, as it was in past studies of TOH problem solving (e.g., Anzai & Simon, 1979). However, it is justifiable – even desirable – from the perspective of contemporary studies, which strive to document the basic information processing of TOH problem solving. For this reason, many of the studies reviewed below taught participants particular strategies. However, it should be noted that allowing participants to induce their own solution strategies introduces less error variation than one might suppose. Participants do not appear to formulate idiosyncratic, inefficient, or errorful strategies (Goel & Grafman, 1995; Kotovsky et al., 1985). Rather, after solving a handful of practice problems, most induce one of the two perceptual strategies.

The fifth dimension of variation is the temporal measure (or measures) reported. There are two. The first is time per individual move. It is typically collected using a constrained presentation paradigm, which forces all student to make the same sequence of optimal moves. The second temporal measure, overall solution time, is the total time required to solve a problem. Time per individual move is obviously the richer of the two temporal measures, revealing the modulation of computational demands within a problem. By contrast, the overall solution time only reveals modulation across problems.

The sixth dimension on which TOH studies vary is the error measure (or measures) reported. There are three. The first is the number of moves required to solve a problem. This measure can only be collected when an unconstrained presentation paradigm is employed because only then can suboptimal moves be made, and can solution sequences deviate from the minimum length. The second measure is error rate, or the proportion of time participants make an error on individual moves. It can only be collected when a constrained presentation paradigm is employed because only then can individual moves be unambiguously classified as errors. (If the presentation paradigm is unconstrained, then participants are free to wander off the optimal solution sequence, and therefore no move after the initial errorful move can be definitively classified as optimal or errorful.) The third error measure is the proportion of problems solved in a fixed amount of time. It is typically collected in studies of patient problem solving for the same reason that many neuropsychological tests enforce deadlines: to avoid interminable response times.

The seventh dimension on which recent studies of TOH problem solving vary is the brain region (or regions) of interest. Although most studies are silent in this regard, there are two exceptions. The first is the set of neuropsychological studies of lesion patients. These studies differ in the granularity with which lesions are classified. Some draw only the coarsest of distinctions, between anterior lesions (i.e., to the frontal lobe) and posterior lesions (i.e., to parietal, temporal, or occipital lobes), whereas others include the orthogonal dimension of lesion laterality. Such neuropsychological studies support indirect inferences from the cognitive impairments of patients to the functions of the damaged brain regions. The second exception is the set of neuroimaging studies of intact normals. These studies directly reveal the neural bases of TOH problem solving with a spatial resolution measured in tens of cubic millimeters.

Moreover, because fMRI and PET can be used with intact normals, neuroimaging studies circumvent the controversies that surround the interpretation of patient data. fMRI studies of TOH problem solving have targeted a variety of brain areas, including dorsolateral prefrontal cortex (DLPFC) and parietal cortex.

The eighth dimension on which contemporary studies of TOH problem solving vary applies only to neuroimaging studies. It is the design employed. Two are of interest. In studies that employ a block design, each block spans the solution of multiple problems of the same kind, and comparisons are made between the average activations of different blocks. For example, there can be blocks of relatively easy problems (e.g., 3-disk problems requiring four moves to solve) and blocks of relatively hard problems (e.g., 4-disk problems requiring eight moves to solve). Comparing the average activation observed during easy versus hard blocks reveals the brain areas sensitive to increasing problem difficulty. In studies that employ event-related designs, multiple images are acquired at a relatively rapid rate (e.g., every 1.5 sec) during the solution of problems. These studies document the ebb and flow of activation over the course of problem solving, which can be compared with the ebb and flow of information processing in computational models. Generally speaking, studies that employ event-related designs provide richer data than studies that employ block designs.

Behavioral Studies of Normal Adults

Most empirical investigations of TOH problem solving have studied normal adults and collected behavioral measures of performance.

Ruiz (1987)

Ruiz (1987) instructed participants in the goal recursion strategy. They then solved 5-disk tower-to-tower problems using a software interface that implemented a constrained presentation paradigm. To ensure that participants actually used the goal recursion strategy, the software forced them to explicitly indicate the establishment of new goals and dis-establishment of satisfied goals. The temporal measure was time per individual move, collected for each of the 31 moves of the optimal solution sequence. The data are shown in Figure 2.

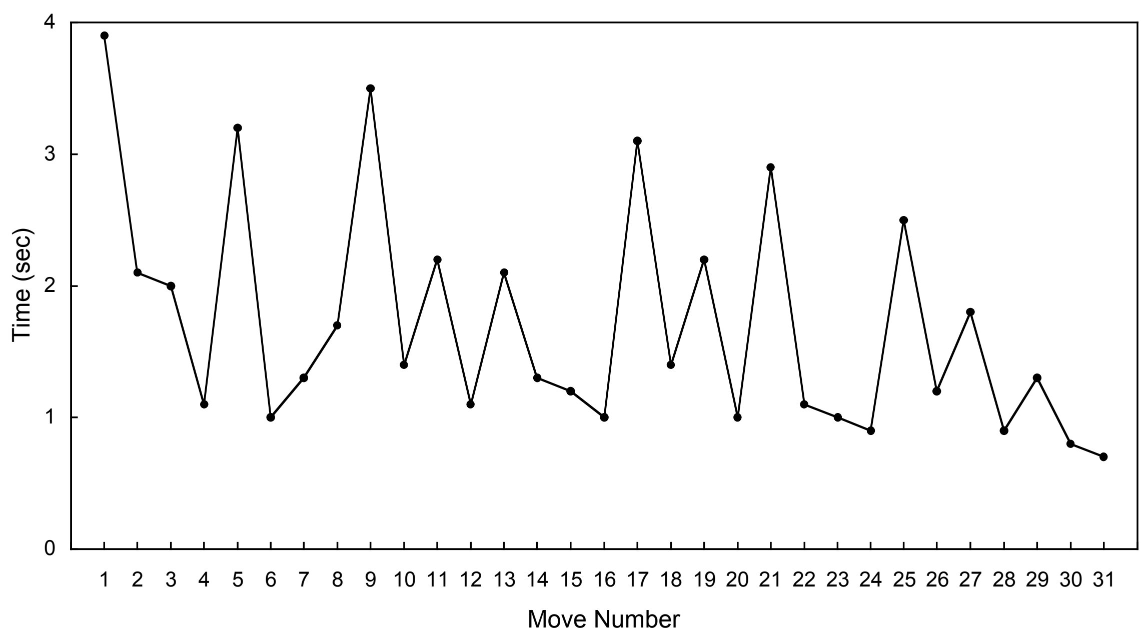


Figure 2. Individual move times for the 5-disk problems of Ruiz (1987).

The key variable was the number of new goals generated before making a move. The major finding was that the time to make a move was an increasing function of the number of new goals. This is shown more clearly in Table 1, which collapses the individual move times across moves requiring generation of the same number of new goals.

Table 1: Average individual move times for Ruiz (1987).

New Goals	Average Move Time
0	1.17 sec
1	1.63
2	2.68
3	3.3
4	3.9

Anderson, Kushmerick, and Lebiere (1993)

Anderson et al. (1993) identified a number of flaws with the Ruiz (1987) study. One was that because participants solved only 5-disk tower-to-tower problems, it is unclear whether the results generalize to problems with different numbers of disks and different kinds of starting and ending configurations. Another flaw was limited ecological validity owing to the constrained presentation paradigm. They conducted a study that fixed these (and other) flaws. A greater variety of problems were used: eight 4-disk problems and eight 5-disk problems drawn from the tower-to-tower, tower-to-spread, spread-to-tower, and spread-to-spread classes. All 4-disk problems required the maximum of 15 moves to solve, all 5-disk problems the maximum of 31 moves.[1] The presentation paradigm was unconstrained. Participants read a one-page description of the goal recursion strategy and were encouraged to use it.

Anderson et al. (1993) collected one temporal measure, overall solution time, and one error measure, number of moves, for each of the 16 problems.[2] These data are listed in Table 2. An interesting feature of the Anderson et al. (1993) data is that problems with spread-ending configurations appear to require more time to solve than problems with tower-ending configurations. This asymmetry has been observed in other studies (e.g., Lehto, 1996). It is indirect evidence that participants employ perceptual strategies, which are sensitive to the visual appearance of puzzle configurations, rather than the goal recursion strategy, which is not.

Table 2: Overall solution time and number of moves for Anderson et al. (1993).

Problem	Overall Solution Time	Number of Moves
4-Disk		
Tower-to-Tower 1	63.34 sec	21.4
Tower-to-Tower 2	51.88	16.9
Tower-to-Flat 1	67.36	19.3
Tower-to-Flat 2	82.29	21.1
Flat-to-Tower 1	68.26	21.2
Flat-to-Tower 2	56.95	17.0
Flat-to-Flat 1	55.13	18.5
Flat-to-Flat 2	80.04	18.4
5-Disk		
Tower-to-Tower 1	178.39	55.4
Tower-to-Tower 2	185.36	54.2
Tower-to-Flat 1	183.54	53.2
Tower-to-Flat 2	216.09	49.0
Flat-to-Tower 1	213.52	58.2
Flat-to-Tower 2	185.20	50.6
Flat-to-Flat 1	237.51	58.5
Flat-to-Flat 2	185.17	48.6

Anderson et al. (1993) also sifted the problem solving traces for instances where problems were solved optimally. These kind of data are normally available only when a constrained presentation paradigm is used. For just these instances, individual moves times were computed; these are plotted in Figure 3 for the 4-disk problems and Figure 4 for the 5-disk data. These data show the same pattern as that of Ruiz (1987): the time to make a move increases with the number of new goals to be generated.

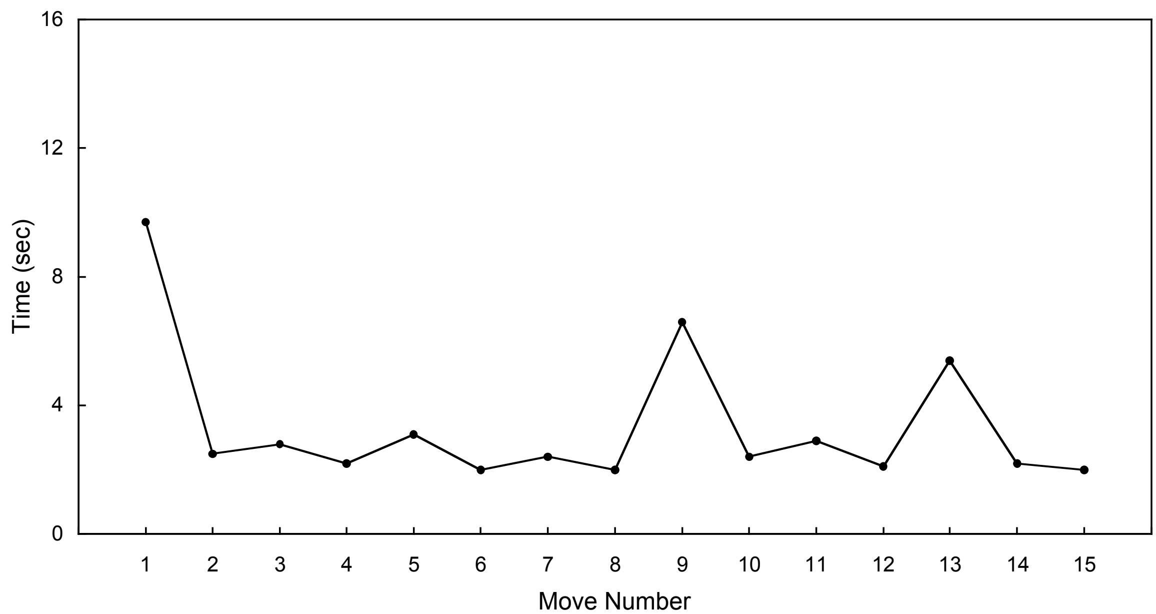


Figure 3. Individual move times for the 4-disk problems of Anderson et al. (1993).

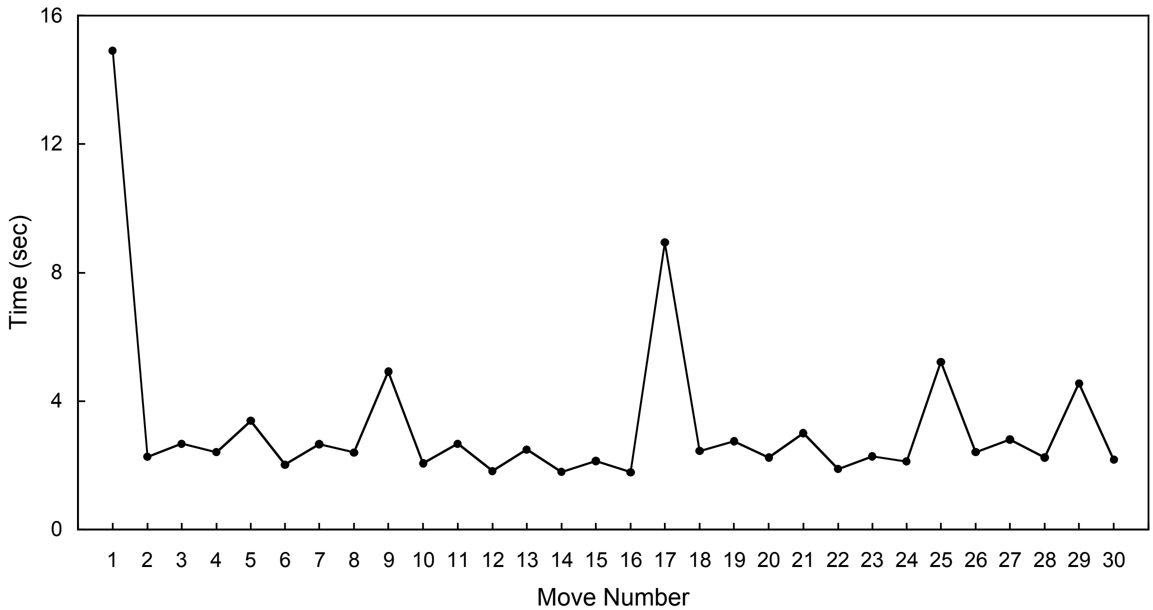


Figure 4. Individual move times for the 5-disk problems of Anderson et al. (1993).

This appearance can be sharpened by collapsing the individual moves times across moves requiring the same number of new goals to be generated. The collapsed data are listed in Table 3. Notice the positive linear relation between the number of new goals generated and the average move time.

Table 3: Average individual move times for Anderson et al. (1993).

	Average Move Time	
New Goals	4-Disk	5-Disk
0	2.14 sec	2.14 sec
1	3.38	2.82
2	4.85	3.53
3	9.7	6.94
4		14.92

Carpenter, Just, and Shell (1990)

Carpenter, Just, and Shell (1990) administered a psychometric test, Ravens Progressive Matrices, to their participants, and stratified them into four performance groups. Ravens is perhaps *the* test of fluid intelligence, which is thought to be at the heart of working memory and executive function – two cognitive constructs that have been localized in part to prefrontal areas (Baddeley, 2003; Curtis & D'Esposito, 2003; Duncan, Seitz, Kolodny, Bor, Herzog, Ahmed, Newell, & Emslie, 2000; Gray, Chabris, & Braver, 2003; Wood & Grafman, 2003). Participants were trained on the goal recursion strategy and then solved a sequence of tower-to-tower problems, beginning with 3-disk problems and progressing up to 8-disk problems. (The 8-disk problems proved too difficult for many participants, and these data were discarded.) The software interface implemented a constrained presentation paradigm.

A single error measure, error rate, was collected; the data are shown in Figure 5. The probability of making an error on a move increased with the number of new goals the move required. There was also an interaction between Ravens score and number of goals: All four groups exhibiting comparable error rates on moves requiring the generation of relatively few new goals. However, lower-scoring groups made errors more frequently than higher-scoring groups on moves requiring the generation of new goals.

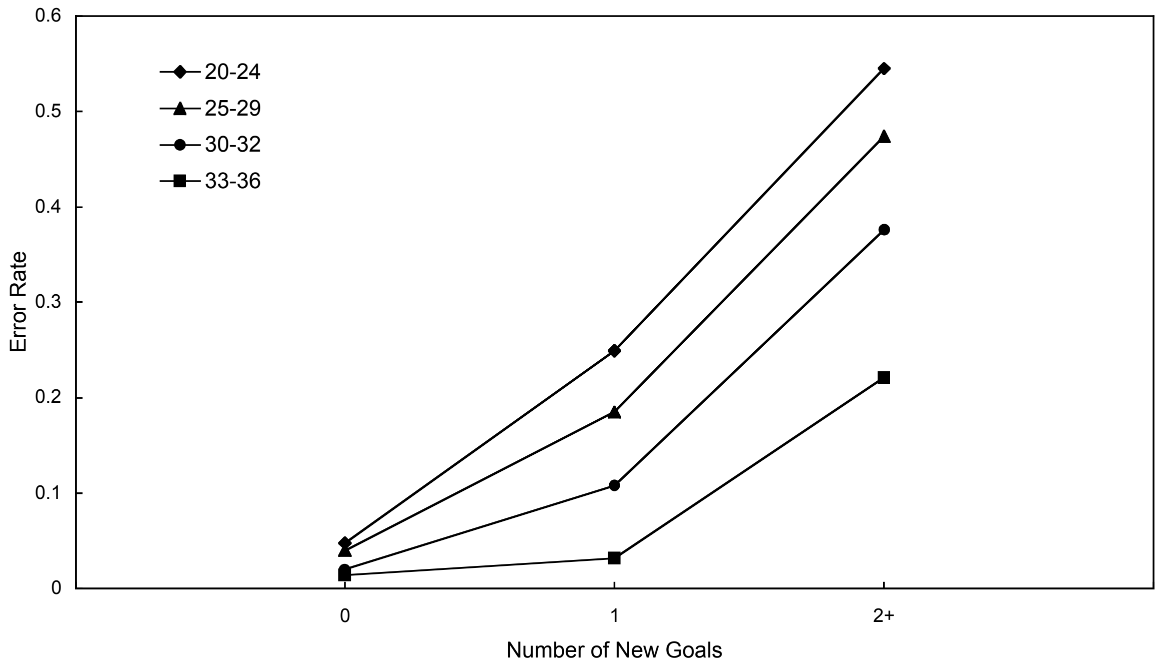


Figure 5. Error rates for Carpenter et al. (1990).

Behavioral Studies of Patients with Frontal Lobe Lesions

The classic laboratory studies of the 1970s focused on the TOH problem solving of intact, normal young adults. Since that time, the TOH task has been adopted by neuropsychologists to probe the cognitive deficits of patients with frontal lobe lesions. These studies have employed behavioral measures exclusively. They are reviewed below.

Goel and Grafman (1995)

The Goel and Grafman (1995) study employed patients with lesions to left prefrontal cortex (PFC), lesions to right PFC, and bilateral lesions to this area, as well as a group of intact, normal controls. Following two practice trials of an unspecified nature, participants solved nine 5-disk spread-to-tower problems, where the tower in the ending configuration was always on the middle peg. These were effectively 4-disk problems because the largest disk was already on the middle peg in all starting configurations. The presentation paradigm was unconstrained, and no strategy instruction was provided.

A number of measures were collected. The temporal measure was overall solution time. One error measure was number of moves, another the proportion of times a problem was solved in the allotted two minutes. These measures were combined into a single, composite score. One result was that the scores of all groups decreased with increasing problem difficulty (roughly length of the optimal solution sequence). Another result was that normal controls scored higher than the patient groups, which were statistically indistinguishable.

Goel, Pullara, and Grafman (2001) revisited the Goel and Grafman (1995) data, presenting them in a format that will be adopted here. Because there were no reliable differences between the three patient groups, they were collapsed into a single frontal group. Offsetting this loss of information was the decomposition of the score measures back into three of its component measures: proportion of problems solved in the allotted time, number of moves, and overall solution time. These measures are plotted in Figures 6, 7, and 8 for both the frontal patients and for normal controls.

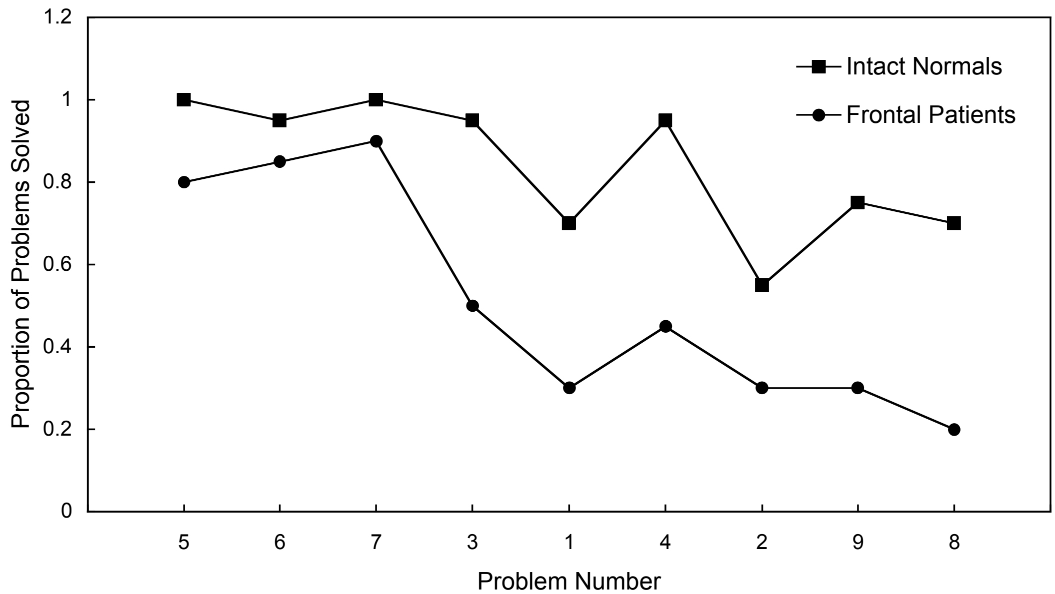


Figure 6. Proportion of problems solved in the allotted time (two minutes) for Goel et al. (2001).

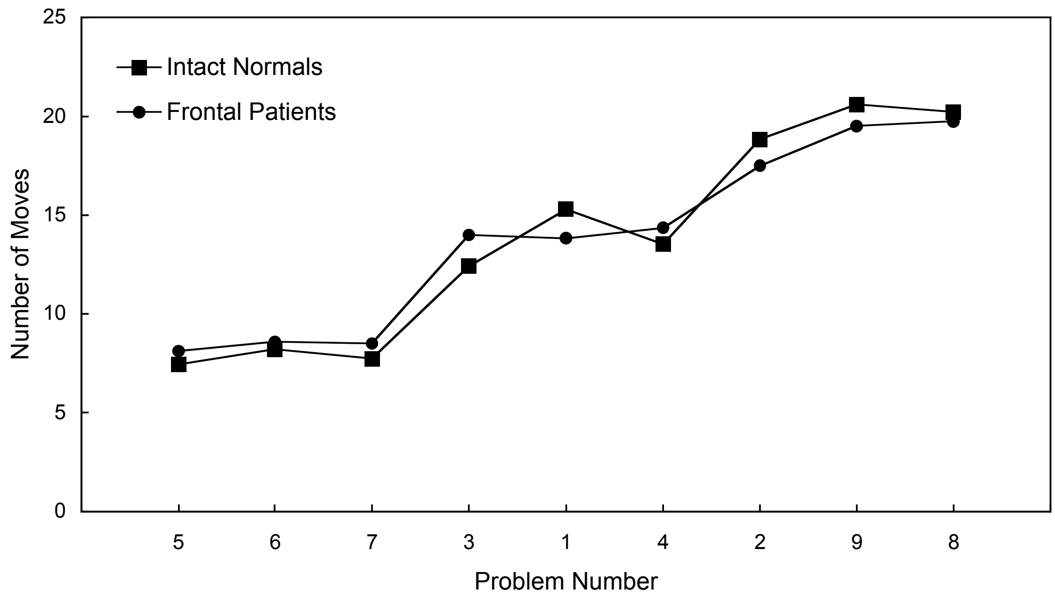


Figure 7. Number of moves required for Goel et al. (2001).

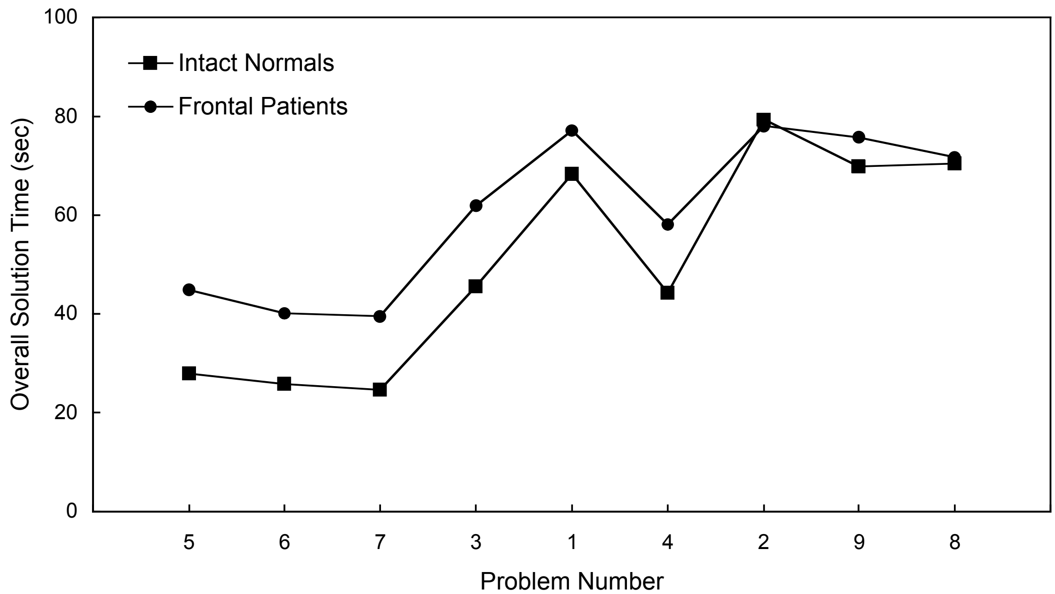


Figure 8. Overall solution time for Goel et al. (2001).

An analysis of individual solution sequences indicated that approximately 80% of normal controls and frontal patients used Simon's (1975) simple perceptual strategy. This strategy produces near-optimal move sequences on problems with three or fewer disks, but is decidedly suboptimal on problems with four or more disks. Such problems can be optimally solved with the sophisticated perceptual strategy, which uses goals to organize the recursive movement of blocking disks. The discrepancy between the simple and sophisticated strategies is particularly large under conditions of goal-subgoal conflict, i.e., when the optimal move from a strategic sense differs from the most perceptually salient one.

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997a)

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997a) tested patients with left frontal lesions, patients with right frontal lesions, and a group of normal controls.[3] (The frontal lesions were to dorsolateral and orbital areas.) 3-disk, spread-to-spread problems were used. Participants solved eight problems, half requiring 4 moves and half 5 moves. Varying orthogonally with solution length was whether problems induced goal-subgoal conflict (Goel & Grafman, 1995), i.e., required a goal-driven rather than perceptually-driven first move; these were the conflict and congruent conditions, respectively. The presentation paradigm was unconstrained, and no strategy instruction was provided.

One error measure, number of moves, was collected. Left frontal patients were impaired on the 4-move conflict problems relative to right frontal patients and normal controls. Estimates of the data are plotted in Figure 9.[4]

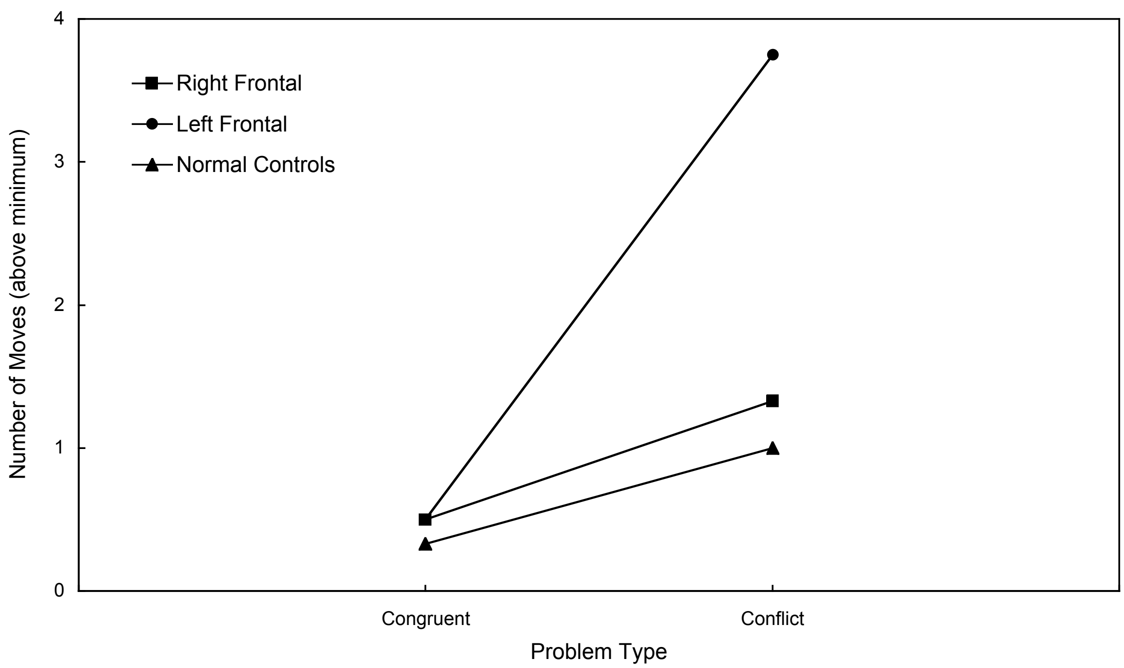


Figure 9. Number of moves (above minimum) required for Morris et al. (1997a). **ESTIMATED**

Two temporal measures were collected. The first, planning time, is defined as the time to make the first move. The planning times for the left frontal, right frontal, and intact normal groups are listed in Table 4. There was a main effect of length, with 5-move problems requiring more planning time than 4-move problems. There was an effect of goal-subgoal conflict for just the 4-move problems, with conflict problems requiring more planning time than congruent problems. The second temporal measure, execution time, is the time to perform all moves but the first one. The execution times for the three groups are also listed in Table 4. The only reliable effect was for the 4-move problems, where conflict problems required longer execution times than congruent problems.

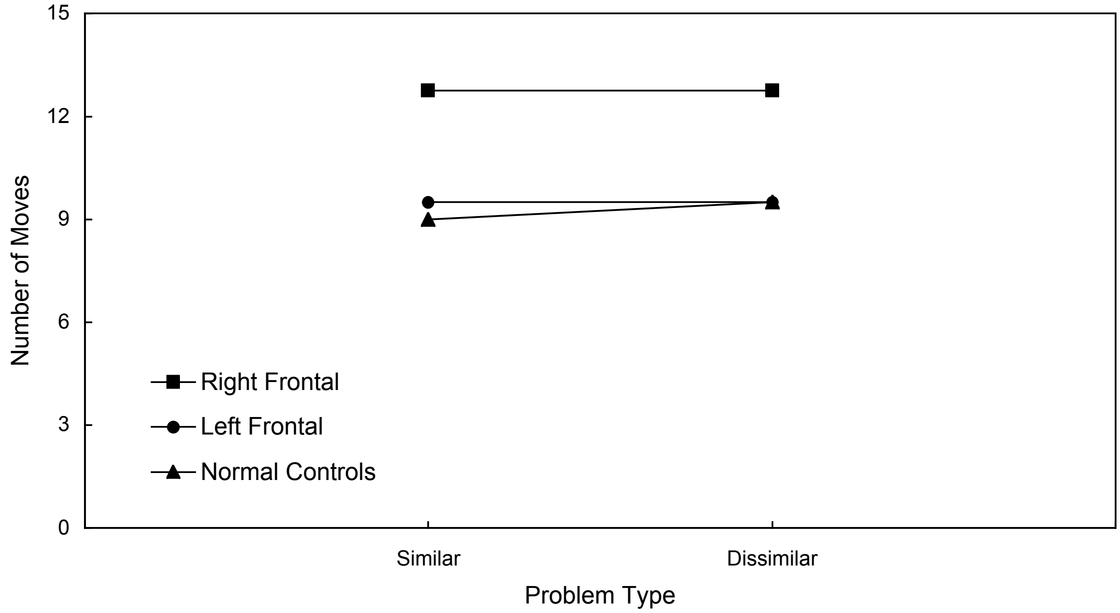
Table 4: Planning times and execution times for Morris et al. (1997a).

	Planning Time		Execution Time	
	Congruent	Conflict	Congruent	Conflict
4-Moves				
Left Frontal	5.98 (6.57)	11.69 (12.33)	3.62 (7.83)	42.10 (46.73)
Right Frontal	3.08 (2.91)	9.25 (9.68)	7.12 (4.89)	26.70 (49.21)
Control	3.99 (2.91)	8.42 (5.77)	8.29 (4.53)	18.00 (27.33)
5-Moves				
Left Frontal	12.28 (10.34)	11.02 (9.05)	17.58 (34.31)	10.97 (13.86)
Right Frontal	8.18 (12.11)	7.48 (8.44)	14.93 (11.44)	18.76 (38.83)
Control	8.72 (5.51)	8.07 (6.10)	8.72 (5.51)	8.07 (6.10)

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997b)

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997b) ran a second study, apparently using the same participants as Morris et al. (1997a). There was a group of patients with left frontal lesions, a group with right frontal lesions, and a group of intact normals.[5] 3-disk problems drawn from the spread-to-spread and spread-to-tower classes were used. Participants solved eight problems that differed on two factors: length of the optimal solution sequence (6 or 7 moves) and whether the two shortest solution sequences were similar in length (i.e., differed by 1 move) or dissimilar in length (i.e., differed by 3 moves). The presentation paradigm was unconstrained, and no strategy instruction was provided.

One error measure, number of moves, was collected. The only effect was one of group: right frontal patients made more moves than the left frontal patients, who were comparable to the normal controls. Estimates of the data are plotted in Figure 10.[6]



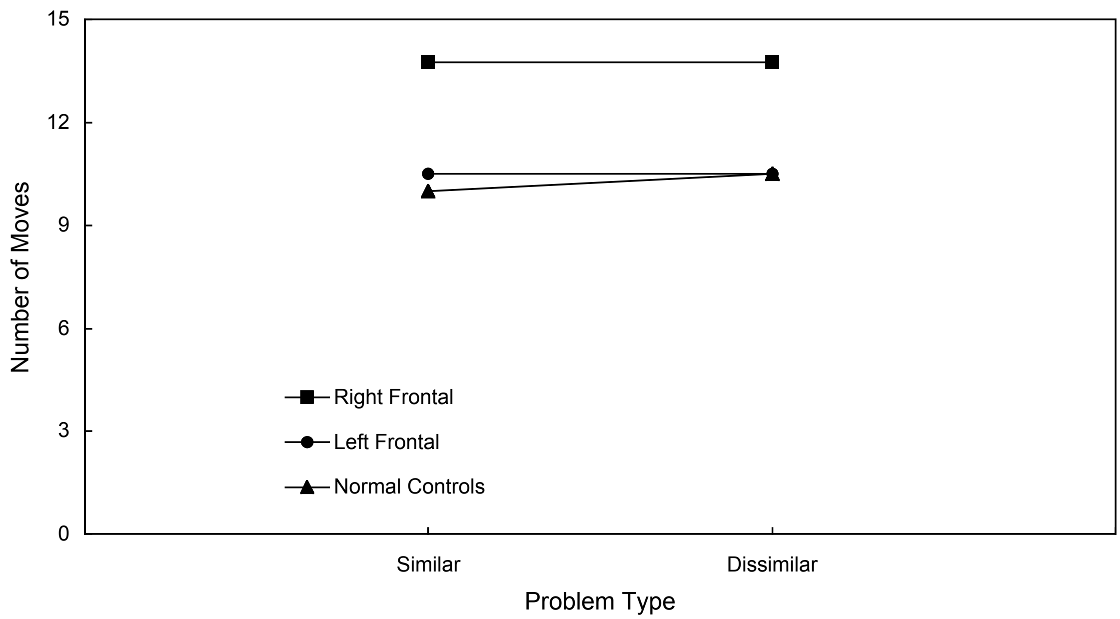


Figure 10. (a) Number of moves required for the 6-move problems of Morris et al. (1997b). **ESTIMATED**
(b) Number of moves required for the 7-move problems of Morris et al. (1997b). **ESTIMATED**

Morris et al. (1997b) collected the same two temporal measures as Morris et al. (1997b), planning time and execution time. The planning times are listed in Table 5. There were no reliable effects, although it appears that right frontal patients required more planning time than left frontal patients and normal controls.[7] The execution times are also listed in Table 5. The left frontal patients produced longer execution times than normal controls on both similar-length and dissimilar-length problems. Right frontal patients produced longer execution times than normal controls on just the dissimilar-length problems.

Table 5: Planning times and execution times for Morris et al. (1997b).

	Planning Time		Execution Time	
	Similar Length	Dissimilar Length	Similar Length	Dissimilar Length
Left Frontal	8.33 (2.04)	7.66 (1.54)	48.15 (14.94)	41.26 (12.58)
Right Frontal	17.74 (6.72)	17.63 (6.72)	25.89 (5.32)	24.43 (8.26)
Control	3.99 (0.44)	8.42 (0.87)	19.15 (2.22)	18.61 (2.15)

Neuroimaging Studies of Normal Adults

Two relevant neuroimaging studies appear in the literature. They are important because they reveal the neural bases of TOH problem solving in intact, normal young adults.

Fincham, Carter, van Veen, Stenger, and Anderson (2002)

Fincham et al. (2002) conducted the first neuroimaging study of TOH problem solving. Participants were first instructed on the sophisticated perceptual strategy, which they practiced extensively outside the scanner. The final 10 practice problems shared the same ending configuration, a tower of disks on the right peg, which participants memorized. Participants then solved 12 criterial problems in the scanner. The criterial problems shared the same ending configuration as the final 10 practice problems, but differed in their starting configurations. All problems had 5 disks and required between 19 and 23 moves to solve optimally. Embedded in each criterial problem were two instances of a 3-disk tower-to-tower subproblem; the resulting 7-move sequences provided the data of interest. Event-related fMRI was employed. Participants were restricted to one move every 16 sec, during which four images were acquired – one every four sec – in various regions of interest. A constrained presentation paradigm was used throughout.

A single temporal measure was collected, time per individual move, for the 7 moves of the embedded sub problems. Estimates of these data are plotted in Figure 11.[8] As in the Ruiz (1987) and Anderson et al. (1993) studies, the time to make a move appears to be an increasing function of the number of new goals it requires.

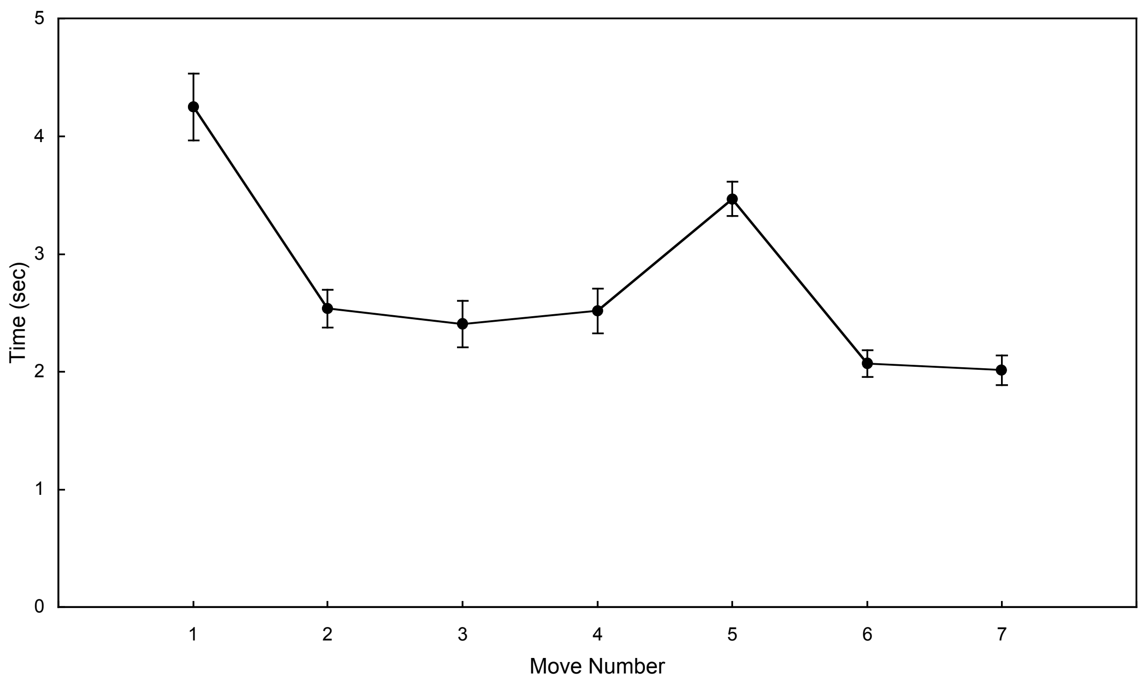


Figure 11. Individual move times for Fincham et al. (2002). **ESTIMATED**

This is shown more clearly in Table 6, which collapses across moves that require the same number of new goals. The longest time is associated with the high-planning move, which requires two new goals. A moderate time is associated with the medium-planning move, which require one new goal. The shortest time is associated with the remaining, low-planning moves, which require no new goals.

Table 6: Average individual move times for Fincham et al. (2002). **ESTIMATED**

New Goals	Average Move Time
0	2.31
1	3.47
2	4.25

A single error measure, error rate, was collected. The average error rate across all 7 moves is 5%, with the highest error rate (11%) associated with the high-planning move.[9]

Neuroimaging data were collected from a number of brain areas, including left DLPFC, right DLPFC, and bilateral parietal cortex. The key result is that the activation in right DLPFC and bilateral parietal cortex is an increasing function of planning load, with high-planning moves producing more activation than low-planning moves. Estimates of the activations observed in right DLPFC are plotted in Figure 12, and estimates of the activations observed in bilateral parietal cortex are plotted in Figure 13.[10] No effect of planning load was found in left DLPFC.

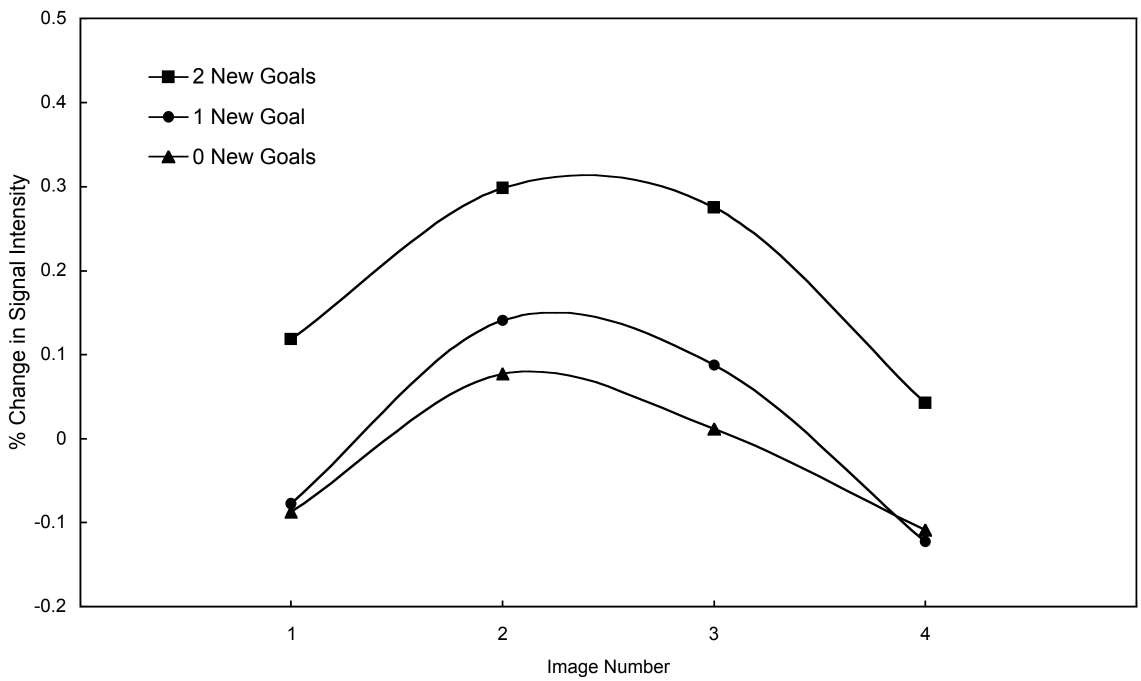


Figure 12. R. DLPFC activations for Fincham et al. (2002). **ESTIMATED**

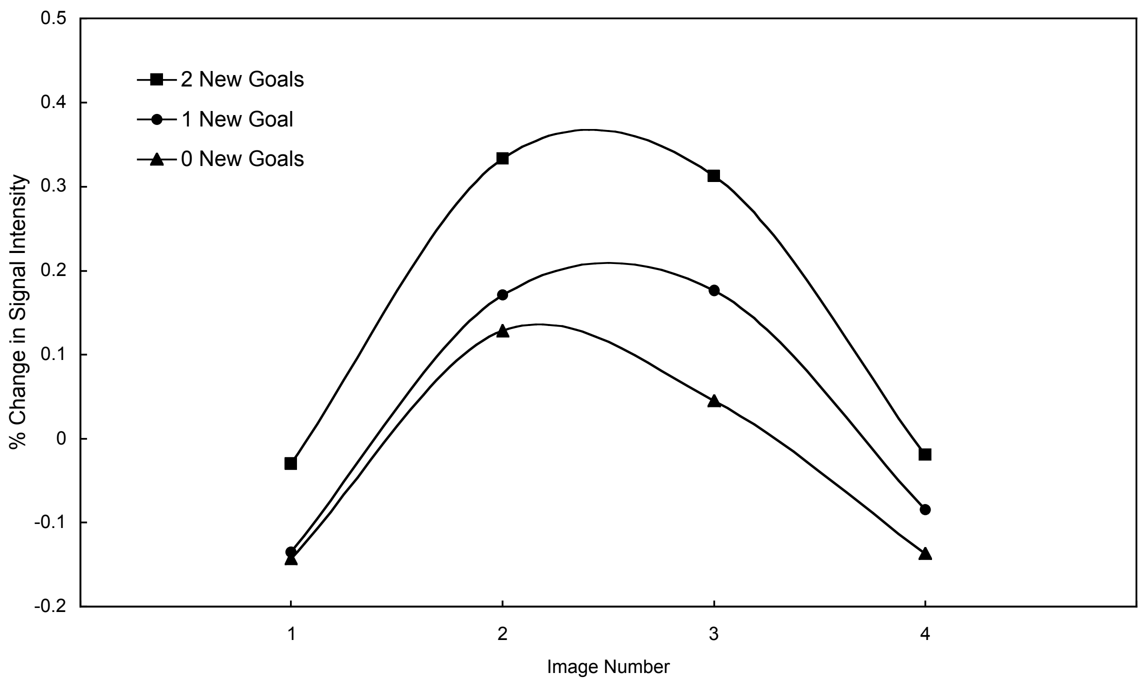


Figure 13. Bilateral parietal activations for Fincham et al. (2002). **ESTIMATED**

Anderson, Albert, and Fincham (2005)

Anderson, Albert, and Fincham (2005) conducted three neuroimaging experiments on TOH problem solving. The focus here is on their first experiment. (The second and third experiments are minor variants.) Anderson et al. (2005) used essentially the same method as Fincham et al. (2002). Participants were instructed on the sophisticated perceptual strategy, which they practiced for two days outside the scanner. The final 10 practice problems shared the same ending configuration. On the third day, they solved 16 criterial problems in the scanner. The criterial problems shared the same ending configuration as the final 10 practice problems but differed in their starting configurations. The problems had 5 disks and required 28 moves to solve optimally. A constrained presentation paradigm was employed. Event-related fMRI was used, with images were acquired every 1.5 sec. The result was an fMRI time series for each of three, broadly-defined regions: a frontal region, a parietal region, and a motor region. (The motor region will be neglected here because the model contains no corresponding center.)

A single temporal measure was collected, time per individual move. Unlike the Fincham et al. (2002) study, this measure was collected for every move, not just for the moves of an embedded subproblem. The data are plotted in Figure 14. Recall that prior studies have found move time to be an increasing function of the number of new goals required. This relationship is also found in the Anderson et al. (2005) data, but it is attenuated.

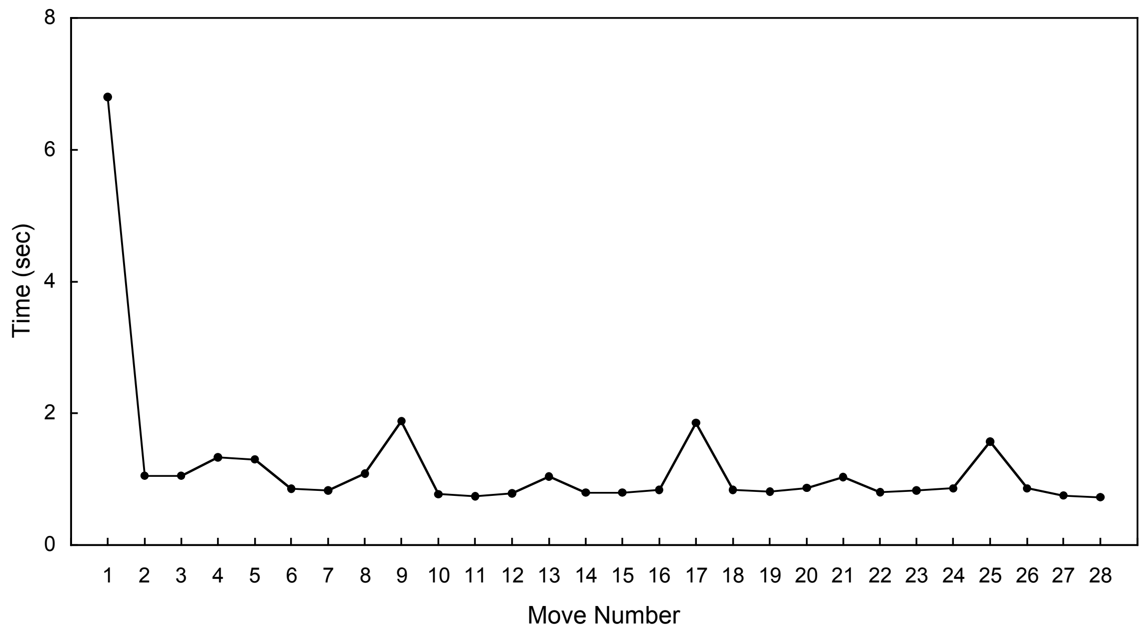


Figure 14. Individual move times for Anderson et al. (2005).

This attenuation is made clear by Table 7, which collapses across moves that require the same number of new goals. Moves times are relatively constant for moves that require two or fewer new goals, but then increase rapidly. Anderson et al. (2005) speculate that this attenuation is a practice effect. Specifically, during their extensive pre-scanner practice, participants learn to chunk the movement of 2-disk and 3-disk towers without the use of goals.

Table 7: Average individual move times for Anderson et al. (2005).

New Goals	Average Move Time
0	0.87 sec
1	1.12
2	1.72
3	1.85
4	6.80

The neuroimaging data for the frontal region and the parietal region are plotted in Figure 15. Anderson et al. (2005) report simple statistical tests on these data: F-tests reveal that the variation in the activation of the parietal region over the final 27 moves of problem solving is reliably different than zero; surprisingly, this is not the case for the frontal region. However, it is unwise to make too much of these simple statistical tests. The neuroimaging data are extraordinarily rich, and must be evaluated against fine-grain theoretical accounts and computational models, as will be done below.

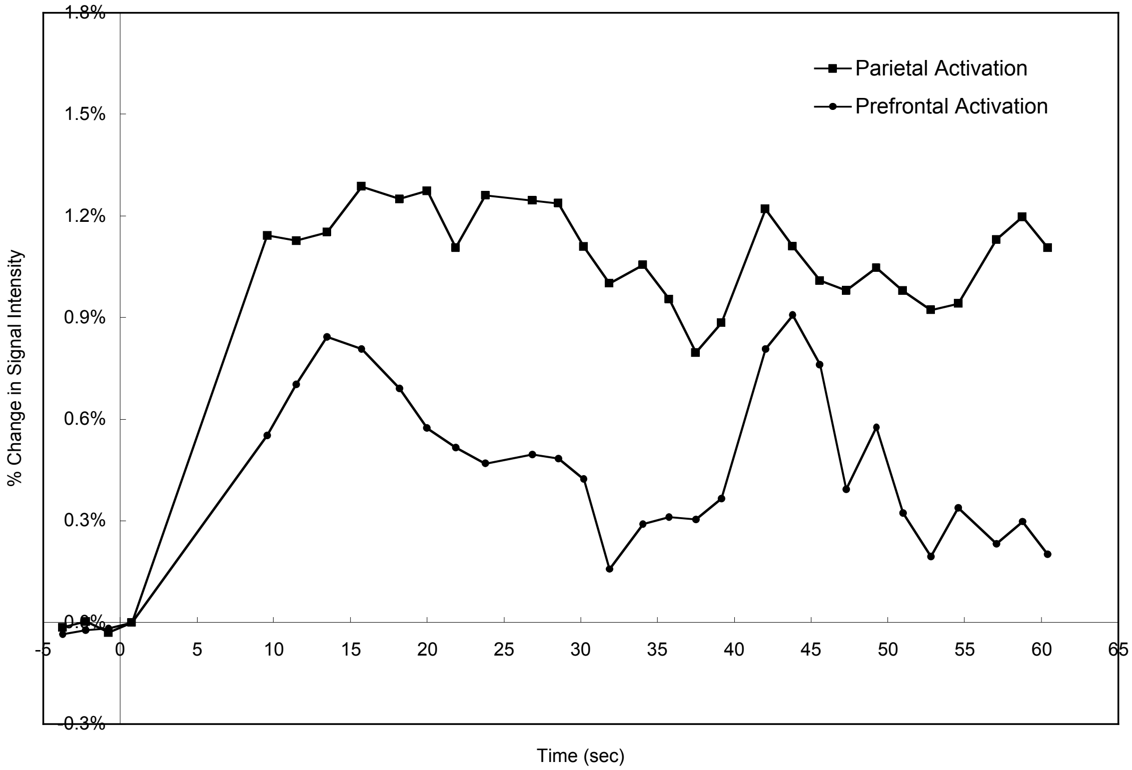


Figure 15. Activation time series for the left frontal and left parietal regions for Anderson et al. (2005).

Chapter IV

EXISTING TOH MODELS

The TOH task has long been used in computer science classes to introduce the notion of recursion (Harel, 1987). It is also a classic toy problem in artificial intelligence for evaluating search and planning algorithms (e.g., Knobloch, 1991). It is therefore surprising that only two models of TOH problem solving were developed during the 1970s, the classic era of research. The UNDERSTAND program of Hayes and Simon (1974, 1977; Simon & Hayes, 1976) explains why different isomorphs of the TOH task vary radically in difficulty. The adaptive production system of Anzai and Simon (1979) simulates the induction of better solution strategies with practice. Although these models addressed the pressing research questions of their time, neither sheds much light on contemporary questions about the information processing demands and neural bases of TOH problem solving.

However, a number of relevant models have appeared over the past decade. Most have been developed within cognitive architectures, which are computational formalisms (i.e., programming languages) tuned by the data of cognitive science. Architecture-based models are less *ad hoc*, and therefore more scientifically credible, than models implemented in general-purpose programming languages such as C++, Java, and Lisp (Newell, 1973). These models are reviewed below, grouped by architectural affiliation.

ACT-R Models

A number of models of TOH problem solving have been developed within the ACT-R cognitive architecture, which combines symbolic and sub-symbolic computational mechanisms. These models share a common focus on the psychological reality of the goal stack. They differ on a number of auxiliary assumptions, including the computational mechanisms that implement the goal stack and the solution strategies they follow.

Anderson, Kushmerick, and Lebiere (1993)

Anderson et al. (1993) developed the first ACT-R model of TOH problem solving. It implements Simon's (1975) goal recursion strategy using three productions: one to recursively decompose the movement of stacks of disks, one to move single disks, and one to suppress satisfied goals. Because the first and second productions are specific to the TOH task, they are assumed to operate 20x slower than the third, more general production, which is presumably closer to the architectural metal. Weakened variants of the first and second productions are also included so that the model sometimes makes suboptimal moves, and thus produces errorful performance. The control structure of ACT-R – its conflict resolution scheme – has been modified so that the choice between competing productions is on the basis of hill-climbing.

Recall that Anderson et al. (1993) collected two temporal measures, time per individual move and overall solution time, and one error measure, number of moves (Table 2 and Figures 3 and 4 in Chapter III). The model was not fit to the data in a quantitative manner. Instead, individual traces were sifted for evidence of goal operations, and this evidence was compared against the model's goal operations in an informal manner.

Anderson and Lebiere (1998)

The Anderson et al. (1993) ACT-R model could not address the temporal data in a quantitative manner because its productions are at the wrong grain size. Each one hides a lot of psychologically interesting computation on its condition side. Anderson and Lebiere (1998) therefore unrolled the three productions into 13 finer-grain productions. For example, some of the new productions perform visuospatial operations, such as the encoding of puzzle configurations and the tracking of disks and pegs, that are opaque tests in the Anderson et al. (1993) model.

The model was fit to Ruiz's (1987) temporal data (Figure 2 in Chapter III) by assuming a default latency of 0.05 sec for ten of the productions and estimating the latencies of the two productions that move disks (1.12 sec each) and the single production that establishes a new goal to unblock the disk to be moved (0.69 sec). The model accounts for 79% of the variance. The model was also fit to the individual move time data that Anderson et al. (1993) collected on 4-disk problems (Figure 3 in Chapter III) by again assuming a default latency of 0.05 sec for most productions and estimating the latencies of the five production that encode different features of the problem being solved (0.56 sec each) and the latency of the two productions that move disks (2.15 sec each). The model accounts for 99% of the variance in the data. Note that Anderson and Lebiere (1998) do not report the model's fit to the individual move time data that Anderson et al. (1993) collected on 5-disk problems (Figure 4 in Chapter III). Also note that because the model is optimal, it cannot account for the error data (number of moves) on the Anderson et al. (1993) problems (Table 2 in Chapter III).

Altmann and Trafton (2002)

Two recent model of TOH problem solving have prompted a rethinking of ACT-R's goal stack. The first is due to Altmann and Trafton (2002), who argue that it is psychologically implausible to endow the human cognitive architecture with a perfect goal stack. They propose instead that goals are stored in declarative long-term memory, and are therefore subject to its errorful properties. The retrievability of a goal is a function of its (1) strengthening through repetition, (2) priming by the environmental cues with which it is associated, and (3) interference with the most-active competing goal. Strengthening, is implemented by goal rehearsal productions and priming by productions conditionalized on visuospatial properties of the current puzzle configuration. Although Altmann and Trafton (2002) describe the model as implementing the goal recursion strategy, it actually appears to use the sophisticated perceptual strategy.

The Altmann and Trafton (2002) model was fit to the temporal data and error data of Anderson et al. (1993). To fit the individual move times, the default values of ACT-R's W, F, d, s, and □parameters were used. The values of the visual encoding time parameter (0.185 sec) were inherited from other ACT-R models of visuospatial tasks (e.g., menu selection) and the value of the movement time parameter (2.15 sec) was taken from Anderson and Lebiere (1998). The model accounted for 99% of the variance for the 4-disk problems (Figure 3 in Chapter III) and 95% of the variance for the 5-disk problems (Figure 4 in Chapter III). These parameter values were then fixed and the model evaluated against the lone error measure, number of moves. Rather than fitting the number of moves required for each of the eight 4-disk problems and each of the eight 5-disk problems, the average number of moves across all 4-disk problems (19.2) and the average number of moves across all 5-disk problems (53.5) were fit. The Altmann and Trafton (2002) model produced similar performance (17.7 and 50.6, respectively).

Importantly, the observed data and model predictions both deviated substantially from the optimal values expected under the assumption of a perfect goal stack (15 and 31, respectively). Note that the Altmann and Trafton model was not fit to the individual move time data that Ruiz (1987) collected on 5-disk problems.

The second model that deconstructs ACT-R's goal stack was developed by Anderson and Douglass (2001). Because it does not differ much from the Altmann and Trafton (2002) model, and because it was only fit to the results of their experiment (and not to the standard data sets on the literature), it will not be discussed further.

Anderson, Albert, and Fincham (2005)

Anderson et al. (2005) adapted the Anderson and Douglass (2001) model to account for their neuroimaging data. This model deconstructs the perfect goal stack of prior versions of ACT-R, decomposing the fundamental operations of pushing and popping into multiple steps. Pushing a goal onto the stack requires (1) formulating a new goal, focusing attention on it, and (2) storing it. Popping a goal from the stack requires (1) retrieving a stored goal and (2) making it the focus of attention. (Another step – encoding the starting and ending configurations – applies only at the start of problem solving, and will not be discussed further.) The model implements the sophisticated perceptual strategy.

The Anderson et al. (2005) model was evaluated against the temporal and neuroimaging data of Anderson et al. (2005). A number of parameters were assigned their default ACT-R values, e.g., each representational operation spans 0.2 sec. Other parameter values were taken from other models, e.g., when making a move, the first finger press requires 0.3 sec and the second 0.2 sec. The remaining parameter values were estimated directly from the data, e.g., memorial operations span 0.22 sec. With respect to their behavioral data, the model accounted for 85% of the variance in the individual moves times (Figure 14 in Chapter III).

To address the neuroimaging data, the latest version of ACT-R claims that the different modules of the underlying production system interpreter correspond to different brain areas. Therefore, computations in these modules can be tabulated in a discrete fashion and convolved with the hemodynamic response function (Aguirre, Zarahn, & D'Esposito, 1998; Boynton, Engel, Glover, & Heeger, 1996) to predict the rise and fall of activation in the corresponding brain areas. When this is done, the model accounts for 62% of the variance in the activation time series observed in the frontal region and 96% of the variance in the activation time series observed in the parietal region (Figure 15 in Chapter III).

3CAPS Models

Two models of TOH problem solving have been developed in 3CAPS, a hybrid cognitive architecture that marries a symbolic production system interpreter with connectionist computational mechanisms such as thresholds, activations, weights, and parallel processing (Just & Carpenter, 1992; Just & Varma, 2002). The key feature of 3CAPS is its constrained working memory, which is realized by limiting the resource available for the storage and processing of representations. By varying the resources available, 3CAPS models can account for a certain class of individual differences in human cognition. The 3CAPS models exploit this capability to account for individual differences in TOH problem solving.

Just, Carpenter, and Hemphill (1996)

The first 3CAPS model of TOH problem solving was developed by Just et al. (1996). It implements the goal recursion strategy of Simon (1975) and is therefore capable of solving tower-to-tower problems. Recall that this strategy works by decomposing larger problems into smaller problems, which are themselves solved. Recursion bottoms out with the movement of individual disks, a primitive operator in the TOH problem space. Each recursive invocation is organized by a goal which must be maintained in working memory. Because working memory resources are limited in 3CAPS, the generation of new goals can draw resources away from existing goals. If the activation of a goal drops below threshold, then it becomes inaccessible to future processing. Inaccessible goals can be recovered by directing activation to them to bring them back above threshold, but this takes time. The Just et al. (1996) model therefore predicts that the more new goals a move generates, the greater the probability that an existing goal will be lost from working memory, and thus the higher the error rate associated with that move.

The model's prediction was tested against the Carpenter et al. (1990) data. Recall that they stratified their participants into four groups based on their scores on the Ravens Progressive Matrices test, a measure of fluid intelligence that correlates highly with measures of working memory capacity. The dependent measure was error rate. They found that error rate increased with the number of new goals to be generated, and that this increase was steeper for participants with lower Ravens scores (Figure 5 in Chapter III). Just et al. (1996) re-grouped the participants into three groups – low, medium, and high – and simulated their performance by endowing the model with a small, medium, or large supply of resources. The simulations displayed the same main effect and interaction seen in the data. The correspondence between the model and data was qualitative; no quantitative measures of fit were reported.

Goel, Pullara, and Grafman (2001)

Goel et al. (2001) developed a 3CAPS model capable of accounting for the TOH problem solving of patients with frontal lobe lesions and intact, normal controls. It implements the sophisticated perceptual strategy. Recall that this strategy uses goals to organize the clearing of blocking disks. As in the Just et al. (1996) model, goals are fueled by a limited resource supply. Therefore, creating new goals can draw resources away from existing goals, rendering them inaccessible in working memory.[11] The Goel et al. (2001) model contains a number of parameters. Some were fixed *a priori*, e.g., when choosing which of two pegs to move a disk to, the correct choice is made 50% of the time. Other parameters were estimated from the normal data and then "lesioned" to account for the patient data. These include the decay rate for the task goal, the decay rate for all other goals, and the latencies of three classes of production: those that move disks, those that do not move disks, and those that recover inaccessible goals.

The model was evaluated against the Goel and Grafman (1995) data. Recall that they combined various component measures into an aggregate score that was plotted separately for patients with left frontal lesions, right frontal lesions, bilateral frontal lesions, and intact normals. Goel et al. (2001) re-organized these data, collapsing the patients (who did not differ reliably in their performance) into a single frontal group, but considering separately the three most informative component measures – proportion of problems solved within two minutes, number of moves, and overall solution time (Figures 6, 7, and 8 in Chapter III). With respect to the intact normals, the model accounts for 58%, 97%, and 77% of the variance of the proportion of problem solved, number of moves, and overall solution time measures, respectively. The model was then lesioned by increasing the decay rate of all goals by 50% and the latencies associated with all productions by 50%. The lesioned model accounts for 73%, 94%, and 74% of the variance associated with the same measures in the frontal patients.

A Connectionist Model

The one connectionist model of TOH problem solving is due to Parks and Cardoso (1997). The model is hard-wired for 3-disk problems. It is a three-layer, feedforward network. The input and output layers each contain 27 units, the hidden layer 15 units. A puzzle configuration is represented by three 3x3 grids, a coarse bitmap of the contents of the left, middle, and right pegs. A grid element has a value of 1 if disk covers it in the visual depiction of the configuration; otherwise, its value is 0. The network was trained by back propagation to solve the standard tower-to-tower 3-disk problem. There was one training pattern for each of the seven moves on the optimal path. Each consists of the configuration before making a move (presented to the input layer) and the one after making the move (used to the compute errors signals at the output units). Additional training patterns were included for every suboptimal (but legal) move from a configuration on the optimal path, with one important exception: the lone suboptimal move from the starting configuration was not included in the training set because it compromised network performance too much. Training reached criterion in 431 iterations.

The model was evaluated using a set of 3584 test patterns representing the seven configurations on the optimal path along with variations of these patterns obtained by adding noise. It makes the optimal move 26% of the time, a suboptimal move 27% of the time, and an illegal move or uninterpretable response 47% of the time. The model was then lesioned by degrading the weights of the connections between the hidden units and output units. The lesioned model was evaluated against the same set of test patterns. It makes the optimal move 17% of the time, a suboptimal move 15% of the time, and an illegal move or uninterpretable response 68% of the time. Parks and Cardoso (1997) note that Glosser and Goodglass (1990) found that intact normals required 10 moves to solve the standard 3-disk tower-to-problem, which optimally requires 7 moves, and patients with left frontal lobe lesions 14 moves. They claim the intact and lesioned models corresponds to these data

The Parks and Cardoso (1997) can be charitably viewed as a first connectionist foray in what has historically been a symbolic domain. Its architecture is hard-wired for standard tower-to-tower 3-disk problems. It was only trained on the neighborhood of patterns along the optimal path, and for many more iterations than participants require to induce solution strategies. It makes the optimal move only one quarter of the time and attempts an order of magnitude more illegal moves than even patients. Finally, the correspondence between the number of moves made by the intact and lesioned models and the intact normals and lesion patients of Glosser and Goodglass (1990) is modest at best, and only informally assessed. It should be noted that there exist more mature connectionist models of Tower of London (TOL) problem solving (Dehaene & Changeux, 1997; Polk, Simen, Lewis, & Freedman, 2002). It is possible that one of these can serve as the starting point for a connectionist model of TOH problem solving that is more competitive with existing ACT-R and 3CAPS models.

Chapter V

THE 4CAPS COGNITIVE ARCHITECTURE

The 4CAPS model of TOH problem solving is developed over the next three chapters. The model consists of three levels, as depicted in Figure 16. This chapter describes its architectural foundation. Chapter VI synthesizes existing theories of problem solving and executive function into a model of fronto-parietal interaction. The instantiation of this model in the TOH domain is described in Chapter VII.

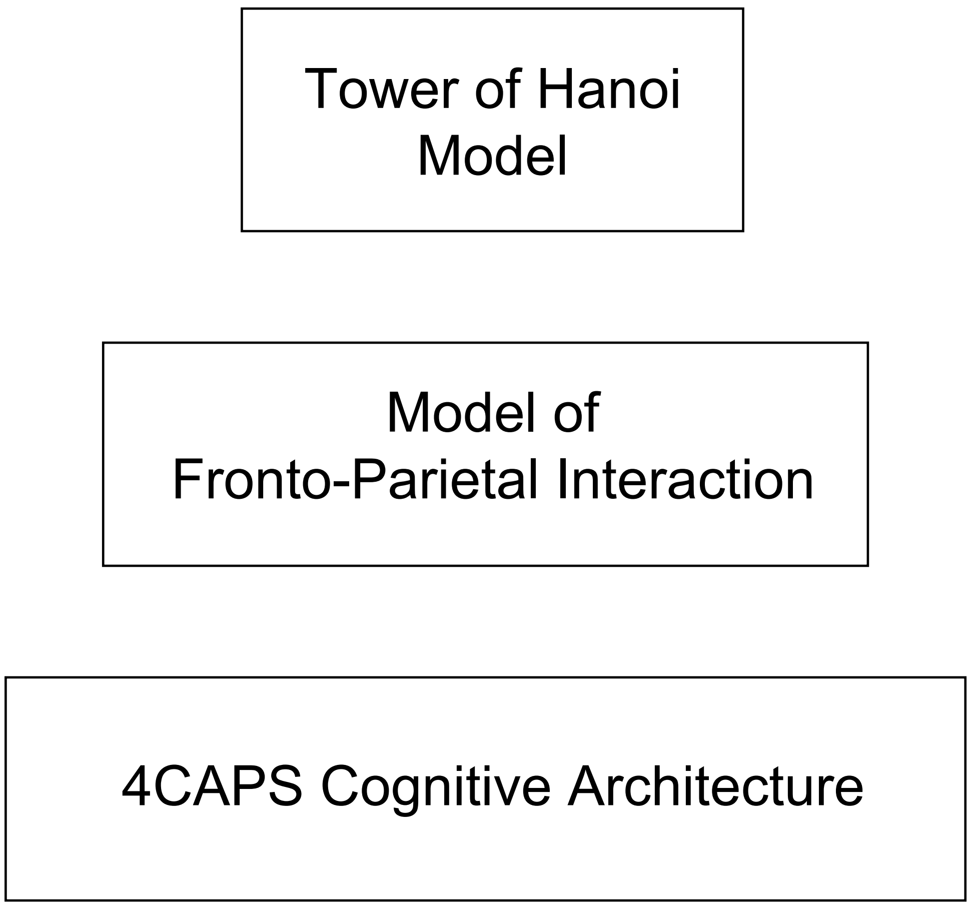


Figure 16. Levels of the 4CAPS TOH model.

Why a Separate Architectural Level? Why 4CAPS?

Computational models are computer programs. They can be expressed in conventional programming languages such as C++, Java, and Common Lisp, but the results often run roughshod over the known characteristics of human information processing (Newell, 1973). For this reason, cognitive scientists have developed custom programming languages that respect the empirical regularities of cognition. Such programming languages are called cognitive architectures.

A number of cognitive architectures are available to cognitive scientists. 4CAPS (Just, Carpenter, & Varma, 1999; Just & Varma, 2006) was chosen for two reasons. First, it targets high-level forms cognition, such as TOH problem solving, and is therefore more appropriate than architectures that target low-level phenomena, such as large-scale neural models (Arbib, Billard, Iacoboni, & Oztop, 2000; Horwitz & Tagamets, 1999) and connectionist networks (e.g., Rumelhart, McClelland, & the PDP Research Group, 1986). Second, 4CAPS is more than a cognitive architectures – it is a cognitive *neuro*architecture. 4CAPS models can account for both behavioral and brain imaging data collected from both intact normals and patients with cortical lesions. In this regard, 4CAPS is superior to other architectures that target high-level cognition, such as EPIC (Meyer & Kieras, 1997) and Soar (Newell, 1990). The only other architecture that addresses both behavioral and brain imaging data collected on high-level cognition is the latest version of ACT-R (Anderson, Bothell, Byrne, Douglass, Lebiere, & Qin, 2004). However, whether ACT-R models can simulate the effects of cortical lesions remains to be demonstrated. (See Just and Varma (2006) for a more detailed comparison between 4CAPS and competing architectures.) For these reasons, the TOH model was developed in 4CAPS.

Although 4CAPS is the most appropriate architecture for the modeling contemporary data on TOH problem solving, it is by no means perfect. 4CAPS and its predecessors are inherently deterministic, and have never produced convincing accounts of error data. As we will see below, the TOH model begins to rectify this deficiency. Another problem with the 4CAPS family of architectures is the absence of learning mechanisms. The TOH model does not overcome this limitation, which remains a topic for future exploration (see Chapter XII).

Historical Development

The original CAPS architecture (Thibadeau, Just, & Carpenter, 1982) synthesizes symbolic and activation-based processing as it was understood in the early 1980s, and in this regard resembles other hybrid efforts of the time (Anderson, 1983; Hofstadter & The Fluid Analogies Research Group, 1983; Holland, Holyoak, Nisbett, & Thagard, 1985; Erman, Hayes-Roth, Lesser, & Reddy, 1980; Minsky, 1985; Rumelhart & McClelland, 1982). Its computational mechanisms include variable-binding, constituent-structured representations, graded activations, weights, thresholds, and parallel processing. The suitability of CAPS for accounting for high-level cognition was demonstrated by successful models of language comprehension (Just & Carpenter, 1987; Thibadeau et al., 1982), mental rotation (Just & Carpenter, 1985), and problem solving (Carpenter et al., 1990).

CAPS was succeeded by 3CAPS (Just & Carpenter, 1992; Just & Varma, 2002), which adds constraints on the resources available for maintaining and processing representations. This enabled computational explorations of individual differences on a number of tasks: sentence comprehension in young adults of different working memory capacities (Just & Carpenter, 1992); sentence comprehension in intact normals and aphasics (Haarmann, Just, & Carpenter, 1997); discourse comprehension in young adults (Goldman & Varma, 1995); problem solving in normal adults different in fluid intelligence (Just et al., 1996); problem solving in intact normals and patients with frontal lobe lesions (Goel et al., 2001); and human-computer interaction (Byrne & Bovair, 1997; Huguenard, Lerch, Junker, Patz, & Kass, 1997). The success of these models furthers the case that human information processing employs hybrid computational mechanisms in a capacity-constrained environment.

CAPS and 3CAPS models account for behavioral measures of high-level cognition collected from normal young adults and neuropsychological patients, broadly defined. 4CAPS, the latest member of the CAPS family, extends to new measures and new populations. Like their predecessors, 4CAPS models account for the time course of cognition and for individual differences. Unlike their predecessors, they also account for neuroimaging measures of normal cognition, and they provide much more precise accounts of the behavioral consequences of cortical lesions. For example, the 4CAPS model of sentence comprehension (Just et al., 1999; Just & Varma, 2006) accounts for reading times at the level of phrases and sentences. It also accounts for the activations observed in Broca's area, Wernicke's area, and their right-hemisphere homologs during sentence comprehension. Finally, because the model makes strong claims about the localization of language function, it can account for the language deficits of patients with perisylvian lesions.

Operating Principles

4CAPS embodies a small number of operating principles that jointly specify cortical and cognitive information processing. Five are relevant for the TOH model.

The first is that thinking is the product of a network of brain areas. A 4CAPS model is not a uniform and monolithic system, but rather a confederacy of centers. Each center maps to a brain area – roughly, a Brodmann area or major gyrus or sulcus – whose information processing it purports to model. At an abstract level, each center is specialized for a number of *cognitive functions*. For example, a center might be specialized for the cognitive function of syntactic parsing. At a concrete level, each center is a hybrid system that combines the computational mechanisms of a production system interpreter with those of a localist connectionist network. Productions encode long-term knowledge. They are graded in their operation, with thresholds on their conditions and weights on their actions. Declarative memory elements encode short-term information; they are akin to the propositions of purely symbolic systems and the feature vectors of connectionist networks. Declarative memory elements are annotated with continuous activation levels.

The specialization of a center for a cognitive function is understood as a claim about the efficiency of resource consumption. More formally, the specialization of center i for cognitive function j is denoted S_{ij}, where $S_{ij} \in [1, \infty)$. A value of 1.0 indicates perfect specialization: 1.0 units of the center's resources are required to perform 1.0 units of the cognitive function. Larger values indicate less-than-perfect specialization, i.e., a specialization of 5.0 means that 5.0 units of resources are required to perform 1.0 units of the cognitive function. If there are N cognitive functions, then the resource consumption of center i can be expressed as:

$$\sum_{j=1}^{N} \left(A_{ij} \times S_{ij} \right)$$

where A_{ij} is the units of cognitive function j performed by center i.

The second operating principle is that each center is endowed with a finite supply of computational resources. This reflects the fundamental fact that there is a bound on the information processing that a finite biological/physical system can perform in a fixed interval of time. Specifically, the resource capacity of center *i* is denoted C_i and the following constraint is enforced:

$$\sum_{j=1}^{N} \left(A_{ij} \cdot S_{ij} \right) \leq C_i \tag{1}$$

Because each center has a finite resource supply, a specialization of ∞ denotes no ability to perform the function.

Third, the resource utilization of each center indexes the biological workload of the corresponding brain area. More formally, the capacity utilization of a center is defined as the proportion of its activation currently being used to store and process representations.

$$CU_i := \frac{\sum_{j=1}^{N} \left(A_{ij} \cdot S_{ij} \right)}{C_i}$$

A major claim of 4CAPS is that the capacity utilization of a center predicts activation in the corresponding brain area.[12] Capacity utilization is an instantaneous index of cognitive workload. It can be averaged over a temporal interval to account for the mean activations observed in block-design studies, and it can be convolved with a hemodynamic response function (Aguirre et al., 1998; Boynton et al., 1996) to account for event-related fMRI data. The proposed mapping between the capacity utilization of a center and the activation of the corresponding brain area makes precise the informal and implicit mapping assumed by those who draw inferences about the localization of cognitive function from neuroimaging data.

The fourth operating principle is that centers collaborate in large-scale networks to perform cognitive tasks. Centers are not modules in the sense of Fodor (1983), specialized for single functions (e.g., syntactic parsing) which they perform in relative isolation from one another (i.e., "information encapsulation"). Rather, each cognitive task can be decomposed into a set of cognitive functions, and recruits a network of centers specialized for their performance. Centers interact frequently, exchanging intermediate representations as they co-articulate the correct task response. For example, in the 4CAPS model of sentence comprehension (Just et al., 1999; Just & Varma, 2006), syntactic parsing is decomposed into multiple cognitive functions. Wernicke's area is specialized for some and Broca's area for others, and therefore both areas are recruited into the large-scale network that comprehends sentences. More generally, no cognitive task is performed by a single center, and no center performs only a single task. Rather, each cognitive task recruits a network of centers, and each center belongs to networks elicited by multiple tasks. This is a coarser form of distributed computation than that found in connectionist architectures (Hinton, McClelland, & Rumelhart, 1986).

Finally, the collaboration patterns between centers are not fixed, but change dynamically. At any point in time, there exists an agenda of pending cognitive functions. Centers with relevant functional specializations are recruited into a large-scale network whose topology reflects local exchanges of information. The membership and topology of the large-scale network change during task performance, the dynamics governed by several factors. One is the logical dependencies that exist between cognitive functions. As pending cognitive functions are performed and new ones scheduled, centers with no-longer-needed specializations exit the large-scale network and centers with now-need specializations enter it. To return to the example of the 4CAPS model of sentence comprehension (Just et al., 1999; Just & Varma, 2006), the syntactic parsing function logically depends on the lexical access function because the outputs of the latter are the inputs of the former. Therefore, syntactic parsing cannot be scheduled until lexical access finishes. Another factor that affects network topology is task complexity. For example, simple

TOL problems can be solved purely by visuospatial processing whereas more difficult problems also require strategic processing. Therefore, solving difficult problems recruits frontal areas known to implement executive functions into the problem solving network (Just & Varma, 2006; Newman, Carpenter, Varma, & Just, 2003). A final factor that affects the topology of large-scale networks is resource availability. A lesion to a brain area is simulated by drastically reducing the resources of the corresponding model center. When a task requires the cognitive functions for which the lesioned center is specialized, then another center must be recruited into the large-scale network to perform them, typically at a lower level of efficiency. Returning once again to the example of sentence comprehension, a lesion to Broca's area prompts recruitment of its right-hemisphere homolog into the language network because it has some specialization for structural processing (Buckner, Corbetta, Schatz, Raichle, & Petersen, 1996; Calvert, Brammer, Morris, Williams, King, & Matthews, 2000; Cao, Vikingstad, George, Johnson, & Welch, 1999; Just & Varma, 2006; Karbe, Thiel, Weber-Luxenburger, Herholz, Kessler, & Heiss, 1998; Thulborn, Carpenter, & Just, 1999)

4CAPS construes the assignment of cognitive functions to centers (or, conversely, the allocation of center resources for the performance of cognitive functions) as a linear programming problem. Such problems can be efficiently solved using the simplex algorithm (Dantzig & Thapa, 1997; Cormen, Leiserson, Rivest, & Stein, 2001). The canonical linear programming problem is to manufacture a number of widgets at a number of factories, where factories vary in their overall production capacities and the efficiency with which each type of widget can be produced. The objective is to maximize the profits (or, equivalently, minimize the costs) of widget production. Widgets map in 4CAPS to cognitive functions and factories to centers. The objective is to assign cognitive functions to centers in a way that maximizes cognitive throughput (or, equivalently, minimizes resource consumption). More precisely, assume that N cognitive functions must be performed by M centers. Recall that A_{ij} denotes the amount of cognitive function j performed by center i; these are the values to be determined, subject to two constraints. First, there are M constraints, one for each center, of form (1) above. They state that each center has a finite resource supply. Second, there are N constraints, one for each cognitive function j, of the form:

$$\sum_{i=1}^{M} A_{ij} \geq R_j \qquad (2)$$

They state that R_j units of cognitive function j must be performed. Assigning cognitive functions to centers requires determining values of the A_{ij} that satisfy the constraints represented by (1) and (2). Because many different assignments exists (e.g., A_{ij}=0 for all i and j), they must be rank-ordered by a measure of goodness. In linear programming problems, this measure takes the form of a linear combination of the A_{ij} to be maximized (or, equivalently, minimized):

$$\sum_{i=1}^{M} \sum_{j=1}^{N} \left(W_{ij} \times A_{ij} \right) \qquad (3)$$

4CAPS defines W_{ij} as $1/S_{ij}$. This ensures that, all other things being equal, cognitive function j will be assigned to the center i most specialized for it.

CHAPTER VI

A MODEL OF FRONTO-PARIETAL INTERACTION

This chapter describes the middle level of the TOH model: a model of fronto-parietal interaction constructed on top of the 4CAPS cognitive architecture (described in Chapter V) that will be specialized for the domain of TOH problem solving (in Chapter VII). Most computational models consist of two levels, an architectural level and a domain-specific level. It is therefore reasonable to ask what is the role of the interspersed middle level? The answer is to unify a number of independent research efforts in cognitive science. The TOH task lies at the intersection of problem solving, executive function, and frontal lobe function. A model of fronto-parietal interaction capable of supporting successful models of the TOH and other tasks has the potential to knit together these separate areas.

Specifically, TOH problem solving is an instance of complex problem solving, an ecologically interesting form of cognition spanning tens of seconds. Little is known about the neural bases of problem solving. By comparison, the neural bases of executive function are relatively well understood. However, executive tasks are typically short in duration (requiring only a few hundred milliseconds to perform), modest in the knowledge they presume, and therefore lacking in ecological interest. The model of fronto-parietal interaction synthesizes existing theories of problem solving and executive function, gaining the respective strengths of each while canceling their weaknesses. It explains an ecologically interesting form of cognition in neurally-localizable terms.

Before proceeding, a word on theory choice is in order. There are many theories of problem solving and there are many theories of executive function. One of each has been selected for synthesis. The claim is not that they are the best theories of their respective domains. Rather, they are a means to a desired end – a model of fronto-parietal interaction capable of explaining TOH problem solving in neurally localizable terms – and must be judged accordingly.

This chapter is organized as follows. First, a prominent theory of problem solving and an equally prominent theory of executive function are described. These theories are then synthesized into a model of fronto-parietal interaction. The model is described informally in this chapter. Its computational details are given in the next chapter alongside those specific to the TOH domain. This commingled organization was chosen because it makes clear the TOH model's operation.

Newell and Simon's Theory of Problem Solving

Over the past half century, Newell, Simon, and their colleagues have articulated a view of problem solving as search through state spaces. Solving a problem requires transforming its starting state into the desired ending state via the application of a sequence of operators. The final version of this theory is Soar (Newell, 1990). Its representations, processes, and control structure are depicted in Figure 17.

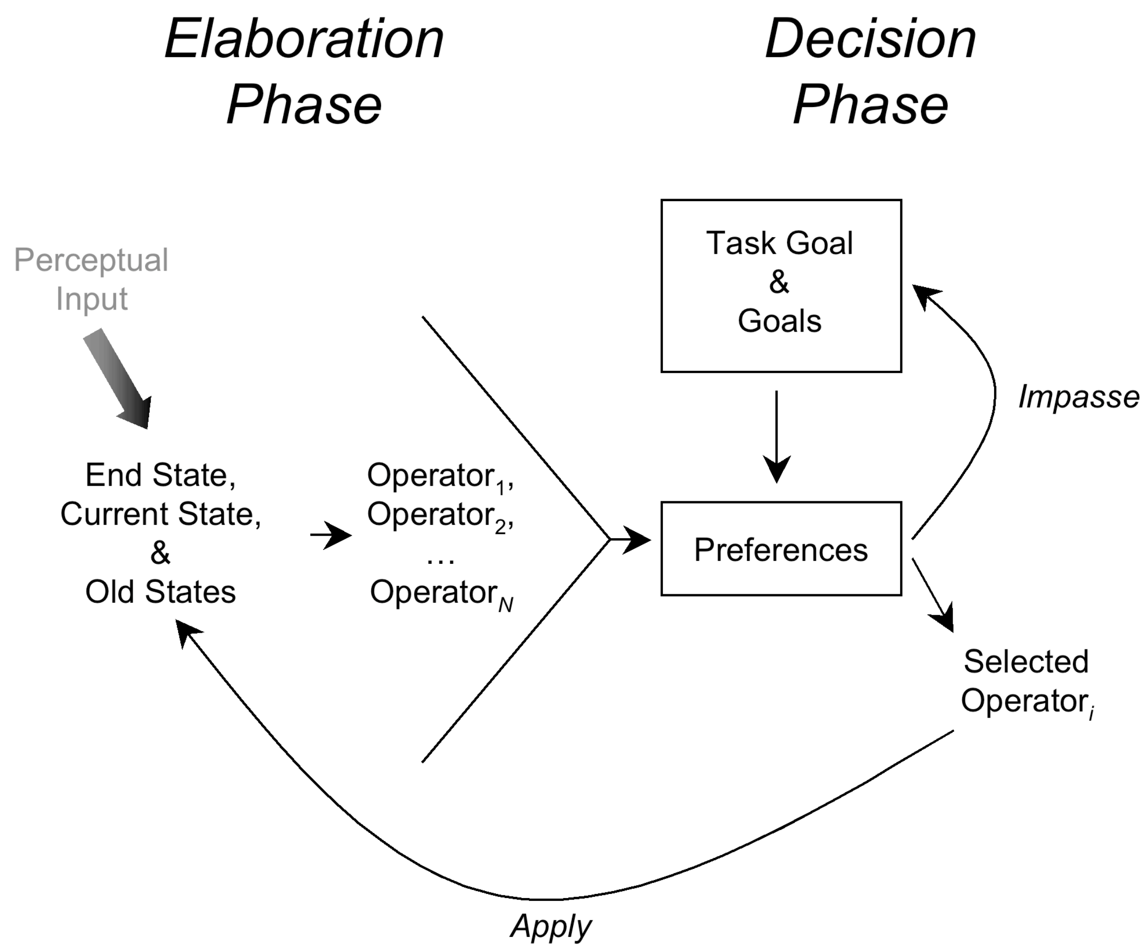


Figure 17. Newell's (1990) Soar theory of problem solving.

The basic unit of time in Soar is the decision cycle, which consists of several phases. During the elaboration phase, operators are proposed that might apply to the current state. This is a sloppy, parallel, and iterative process. During the decision phase, preferences are proposed that assert the relative merit of pairs of operators. At the end of the decision phase, declarative memory contains zero or more operators and zero or more preferences that order these operators. A decision procedure – a computational primitive of Soar – then sorts the operators based on the preferences. If one operator is unambiguously preferred over all others, then it is selected and applied to the current state, producing a new current state. If no operators were proposed during the elaboration phase or if no single operator emerges as most preferred, then an impasse is said to have occurred. A goal is established to resolve the impasse and a new problem space is spawned in which to pursue this resolution. This pursuit can produce additional impasses, which lead to the recursive establishment of goals and the spawning of nested of problem spaces. When an impasse is resolved, the associated goal is popped and the associated problem space dispensed with.

Three aspects of Soar merit further discussion. First, strategic problem solving is driven by goals, of which there are two kinds. The task goal is to solve the problem at hand, which is defined by its starting and ending states. (It defines the "task set," to use different terminology.) The second kind are the goals established at impasses which guide problem solving in nested problem spaces. Another notable aspect of Soar is that its view of problem solving as search through problem spaces is perfectly general – and therefore perfectly useless out of the box. Its generic states, operators, and goals must be specialized to capture the details of the domain being modeled. Finally, Soar includes a powerful learning mechanism, chunking, that learns to avoid the impasses that arise during problem solving. Learning is outside the scope of this dissertation project, as discussed in Chapter XII.

Shallice's Theory of Executive Function

Shallice and his colleagues (Norman & Shallice, 1986; Shallice, 1982) have developed a theory of executive function that has long dominated discussions in cognitive neuropsychology. This theory has also proven influential within cognitive psychology and cognitive neuroscience. For example, Baddeley (1996) endorses it as a possible account of the central executive of his theory of working memory. The components of Shallice's theory and their neural localizations are depicted in Figure 18.

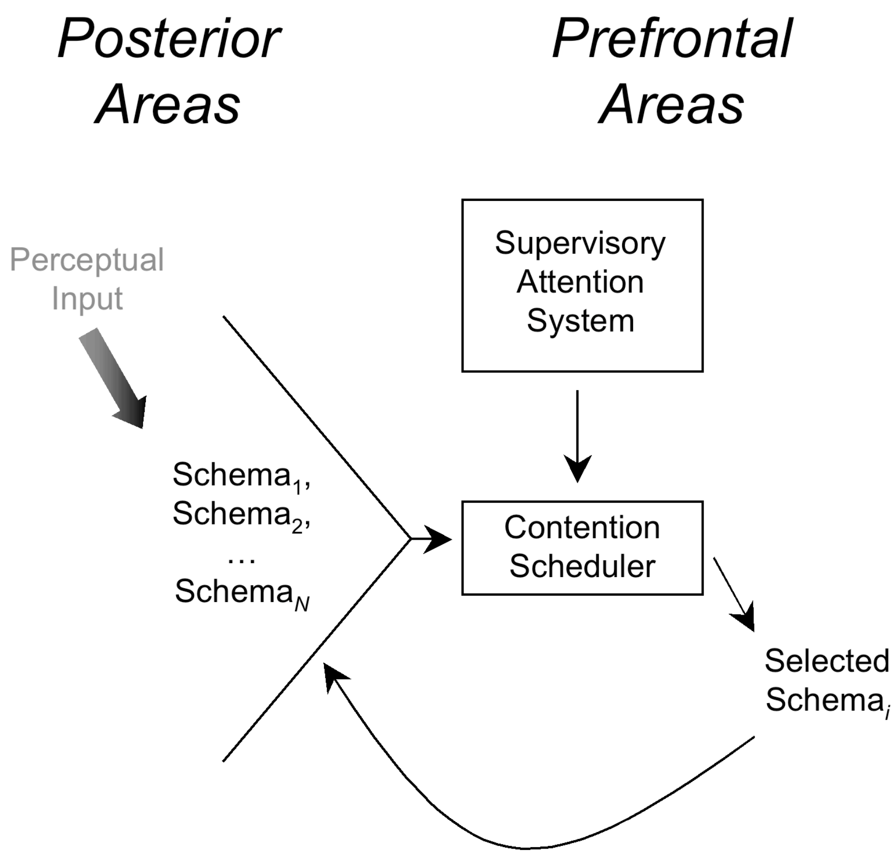


Figure 18. Shallice's (1982) theory of executive function.

The fundamental cognitive unit in Shallice's theory is the schema, a "highly specialized routine program" that controls an overlearned action or skill (p. 199, Shallice, 1982). Schemas are triggered by perceptual inputs and, in more complicated assemblages, by the outputs of other schemas. At any point during task performance, multiple schemas vie for control of cognition. It is the job of the contention scheduler to select between them. Selection is via conventional activation dynamics. Specifically, each schema has an associated activation, and is selected when its activation exceeds some threshold. Selection is said to be routine when a schema gains control by exciting itself and inhibiting its competitors. A routinely selected schema can maintain control over a short period of time, even in the absence of supporting perceptual input, by coordinating the selection of subordinate schemas. When routine scheduling fails, non-routine action is required. This is the purview of the Supervisory Attention System (SAS), which formulates plans that resolve contentions for the control of cognition. These plans do not seize control in a top-down fashion, as in the conventional planning systems of artificial intelligence. Rather, they operate indirectly, biasing the activations of competing schemas, and thus the likelihood they will be selected by the contention scheduler.

The neural localization of schemas, the contention scheduler, and the SAS traces back to the neuropsychological antecedents of Shallice's (1982) theory. Luria (1966) claimed that executive function is the product of two processes, simultaneous synthesis and successive synthesis. Simultaneous synthesis is the parallel processing of incoming stimuli into spatial groupings – schematic processing, to use Shallice's terminology. Luria attributed simultaneous synthesis to occipital and parietal areas. Successive synthesis is the serial processing of temporal sequences – contention scheduling and supervisory attention, in the language of Shallice. Luria mapped successive synthesis to frontal and fronto-temporal areas. Another neuropsychological precursor is Milner's (1982) account of frontal lobe function. She argued that the temporal organization of behavior is lateralized. Left frontal lesions impair selection between competing responses – contention scheduling, to use Shallice's terminology. Right frontal lesions compromise the short-term buffering of information – a function of the SAS, in the language of Shallice's theory.

Shallice himself has consistently localized schemas to posterior areas, but has changed the mappings of the contention scheduler and SAS over the years. Initially, the contention scheduler was mapped in a general way to the frontal lobe and the SAS more specifically to left frontal cortex (Shallice, 1982). Note that this mapping is inconsistent with Milner's (1982) attribution of SAS-like functions to right frontal areas. Norman and Shallice (1986) shifted the localization of the contention scheduler to the corpus striatum of the basal ganglia based on the analogy between the sequencing motor actions, which this area is known to perform, and the sequencing of schemas. The localizations change again when Shallice and Burgess (1996) mapped both the contention scheduler and SAS to prefrontal cortex.

There is presently no consensus on which area (or network of areas) in prefrontal cortex implements the contention scheduler and which the SAS. The localizations proposed below extend those of Luria (1966), Milner (1982), and Shallice and Burgess (1996).

A Theoretical Synthesis

The Soar theory of problem solving and Shallice's theory of executive function have been synthesized into a model of fronto-parietal interaction. This synthesis is summarized in Table 8, and described in greater detail below.

Table 8: Synthesis of the Soar theory of problem solving and Shallice's theory of executive function.

Soar	Shallice's Theory	Proposed Neural Localization
states, direct operators	schemas	intra-parietal sulcus
goals, indirect operators	SAS	right dorsolateral prefrontal cortex
preferences	Contention Scheduler	left dorsolateral prefrontal cortex

Schemas

Shallice has consistently localized schemas to posterior areas. The TOH task is visuospatial in nature, and therefore the relevant posterior areas are in the parietal lobe. (By contrast, if the task was an instance of linguistic problem solving, such as syllogistic reasoning, then the relevant posterior areas might be in the temporal lobe (Goel, 2003).)

Left intra-parietal sulcus (IPS) has been implicated in the storage and transformation of visuospatial representations (Alivisatos & Petrides, 1996; Corbetta & Shulman, 2002; Trojano, Grossi, Linden, Formisano, Goebel, Cirillo, Elefante, & Di Salle, 2002). In the vocabulary of Soar, this area is the repository of states: starting states, ending states, and the intermediate states visited during the search process. Left IPS is also the transformer of states, applying the selected operator to the current state to generate the new current state.

Right IPS has been implicated in visual attention (Corbetta & Shulman, 2002; Ng, Eslinger, Williams, Brammer, Bullmore, Andrew, Suckling, Morris, & Benton, 2000; Trojano et al., 2002). In the vocabulary of Soar, this area proposes new operators. These operators are triggered by differences between perceptual inputs (i.e., the ending state, which is typically available in the external environment) and visuospatial representations (i.e., the current state, which is typically encoded in left IPS). They reduce these differences, and are therefore termed *direct* operators. Right IPS is promiscuous, proposing direct operators with no regard as to whether they are optimal (i.e., the best), applicable (i.e., their preconditions are satisfied), or even legal (i.e., consistent with the task instructions).

The SAS

Shallice describes the SAS as containing the high-level schemas that organize the (low-level) schemas found in posterior areas. These high-level schemas are likened to the scripts of Schank and Abelson (1977) and the MOPs and TOPs of Schank (1982). For example, the restaurant script – a general description of dining out – is found in the SAS. It coordinates the execution of schemas corresponding to primitive dining events such as checking one's coat, ordering dessert, and paying the bill. The high-level schemas of Shallice's theory can be localized in a general way to the frontal lobe (Luria, 1966), specifically to right prefrontal cortex (Milner, 1982; Shallice & Burgess, 1996). In this dissertation project, they are mapped more precisely to right DLPFC.

The fronto-parietal model interaction implements the supervisory control functions of the SAS using the computational mechanisms of Soar. The SAS has three responsibilities. First, it maintains the task goal (or task set), which is to perform the task at hand. Second, when impasses arise during routine action, it establishes goals to resolve these impasses. This is potentially a recursive process capable of articulating the goal hierarchies that guide non-routine action. The third responsibility of the SAS is to propose operators that achieve outstanding goals, i.e., resolve impasses. These operators are *indirect* in that they do not necessarily reduce difference between perceptual inputs (i.e., the ending state) and internal representations (i.e., the current state), as is the case with the direct operators proposed right IPS.

The Contention Scheduler

In Shallice's theory, the contention scheduler is the arena where schemas compete for selection. They do so incrementally using excitatory and inhibitory activation. Selection in Soar is realized by very different computational mechanisms. Specifically, at the end of the elaboration phase, multiple operators potentially exist. Some are direct, triggered by differences between the current and ending states; these are proposed by posterior areas such as right IPS. Others are indirect operators that achieve outstanding goals; these are proposed by an anterior area, right DLPFC. Selection is a two step process. Preferences are first asserted, each indicating the relative goodness of the corresponding operator. Next, the preferences are sifted and the most preferred operator is selected for application. Contention scheduling is localized in a general way to the frontal lobe (Luria, 1966), specifically to an area in the left prefrontal cortex (Milner, 1982; Shallice & Burgess, 1996). In this dissertation project, it is attributed more precisely to left DLPFC.

Whereas Shallice's theory realizes selection via the activation-based process of contention scheduling, Soar employs the discrete, symbolic machinery of preference-based adjudication. Reconciling these two schemes requires finding a compromise between the continuous and the discrete, between the subsymbolic and the symbolic. This is the most difficult step in synthesizing the two theories. Because there is no *a priori* correct way to soften the preferences of Soar with the activation dynamics of Shallice's contention scheduler, several attempts were made. These are described in the next chapter and evaluated empirically in Chapter VIII.

CHAPTER VII

THE TOH MODEL

To review, the 4CAPS TOH model consists of three levels. The bottom level, the 4CAPS cognitive architecture, was described in Chapter V. The middle level, the model of fronto-parietal interaction, was sketched in Chapter VI. This chapter fills in the details of the fronto-parietal model and describes its instantiation in the TOH domain. The emphasis is on the design decisions that underlie the model's construction. A design decision is a choice between alternative implementations, where the choice cannot be made *a priori* because the different implementations have offsetting pros and cons. Design decisions are a computational form of degrees of freedom. This can be confusing because degrees of freedom are commonly thought to be synonymous with the numerically-valued free parameters found in mathematical and statistical models. Computational models typically contain both free parameters and design decisions. The relation between degrees of freedom, free parameters, and design decisions will be taken up again in Chapter XI.

The five design decisions that underlie the 4CAPS model of TOH problem solving are summarized in Table 9. Four are binary-valued and one ternary-valued, yielding (2^4x3=) 48 model variants. The model variants make differentiable predictions that will be evaluated against behavioral and brain imaging measures of normal and patient performance in subsequent chapters. From a computational perspective, the goal is to assess the five design decisions that undergird the model, and to determine the value of each one. From a Popperian perspective, the goal is to identify *the* model variant that the data best corroborate.

Table 9: Design decisions of the model.

Design Decision	Values
preference-scheme	*absolute, relative*
accrual-scheme	*multiplicative, additive*
top-moves	*yes, no*
top-goal-moves-only	*yes, no*
suppress-old-states	*all, non-goal, none*

The organization of this chapter parallels the internal structure of the TOH model, depicted in Figure 19. The model's four centers are first described. The cortical mapping of each is specified and its representations and processes reviewed in some detail. The distinction between the fronto-parietal and TOH levels of the model is given special emphasis throughout, as are the five design decisions. This is a high-level description of the model. An intermediate-level description is presented in Table 10, which lists for each model production the center to which it belongs, the level (fronto-parietal or TOH) it occupies, and the design decisions on which it is conditionalized on (if any). A low-level description – the model's actual code – is presented in Appendix A. This chapter concludes with an account of how the four centers collaborate to solve TOH problems.

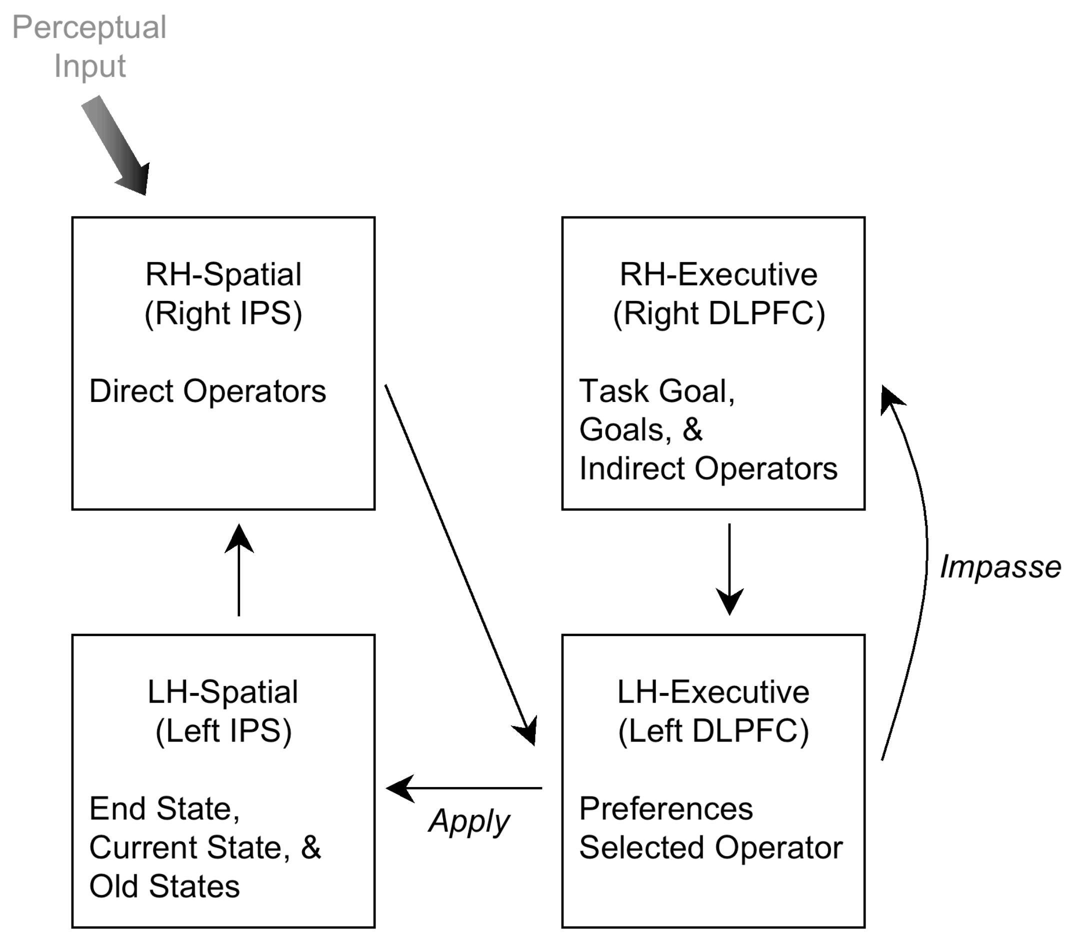


Figure 19. Centers of the 4CAPS TOH model, their functional specializations, and their pattern of collaboration.

Table 10: Productions of the model.

Center	Model Level	Production
LH-Spatial	Fronto-Parietal	suppress-old-state[a]
		suppress-non-goal-state[b]
	TOH	perform-move
RH-Spatial	Fronto-Parietal	suppress-unselected-operator
		suppress-applied-selected-operator
		suppress-illegal-selected-operator
		suppress-unapplied-selected-operator
	TOH	propose-top-move[c]
LH-Executive	Fronto-Parietal	prefer-legal[d, e, f]
		prefer-hill-climbing[e, f]
		prefer-steepest-hill-climbing[e, f]
		prefer-goal[e, f]
		prefer-top-goal[e, f]
		iteratively-activate[g]
		select-among-preferences
		suppress-preference
		suppress-applied-selected-operator-marker
		suppress-illegal-selected-operator-marker
		suppress-unapplied-selected-operator-marker
	TOH	*none*
RH-Executive	Fronto-Parietal	suppress-satisfied-goal
	TOH	propose-unblock-disk-goal
		propose-unblock-position-goal
		suppress-superfluous-unblock-disk-goal
		suppress-superfluous-unblock-position-goal
		propose-unblock-disk-move[d]
		propose-unblock-position-move[d]

[a] Defined when *suppress-old-states* is *all*
[b] Defined when *suppress-old-states* is *non-goal*
[c] Defined when *top-moves* is *yes*
[d] The exact form of this production depends on the value of *top-goal-moves-only*
[e] The exact form of this production depends on the value of *preference-scheme*
[f] The exact form of this production depends on the value of *accrual-scheme*
[g] Defined when *accrual-scheme* is *multiplicative*

The Spatial Centers

In Shallice's theory, domain knowledge is encoded in schemas, which are perceptually-triggered in their operation and localized to posterior areas. The model of fronto-parietal interaction implements schemas using the states and (direct) operators of Soar. These computational mechanisms are localized to parietal areas – specifically, to IPS – reflecting the visuospatial nature of the TOH task and the other tasks to which the fronto-parietal model has been applied: TOL problem solving, mental rotation, and driving.

LH-Spatial

The LH-Spatial center corresponds to left IPS. It is the visuospatial workspace of the fronto-parietal model, the repository and transformer of visuospatial representations. LH-Spatial maintains the current state if it is not visually available in the external environment. It also maintains the ending state on those occasions when it is not visually available in the external environment. Whether LH-Spatial maintains other states is the subject of the design decision described below. The maintenance function attributed to LH-Spatial is domain-general, and therefore belongs to the model's fronto-parietal level. LH-Spatial is also charged with applying the selected operator to the current state to generate a new current state. The details of operator application and state creation are domain-specific, and therefore belong to the model's TOH level.

One design decision impacts LH-Spatial: whether old (i.e., no longer current) states are maintained. This is the *suppress-old-states* decision, and it is ternary-valued. If the value is *all*, then each state is suppressed as soon as it is succeeded by a new state. This has the benefit of minimizing resource consumption in LH-Spatial which, like all centers, possesses a limited resource supply. When this is the case, the following production is defined in LH-Spatial at the fronto-parietal level.

suppress-old-state
If *new-state* is more recent than *old-state*
Then suppress *old-state*

If the value of the *suppress-old-states* design decision is *non-goal*, then all old states not associated with active goals are suppressed. In other words, only old states that resulted in impasses, and thus the creation of new goals, are maintained. When this is the case, the following production is defined in LH-Spatial at the fronto-parietal level.

suppress-non-goal-state
If *new-state* is more recent than *old-state*
there exists no *goal* that resolves an impasse in *old-state*
Then suppress *old-state*

Finally, if the value of the *suppress-old-states* design decision is *none*, then no old states are suppressed. In other words, all old states – the entire search path from the starting state to the ending state – are maintained. This is a resource-intensive decision, but it can be justified if the task goal is, for example, to remember the entire solution path for later reproduction. When this is the case, no additional productions are defined in LH-Spatial.

In addition to its maintenance function, LH-Spatial is also specialized for applying the selected operator to the current state to generate the new current state. This function is implemented by the following production in LH-Spatial.

perform-move
If *configuration* is the current state
move is the selected operator
Then apply *move* to *configuration*

This production is necessarily couched in the language of the TOH task – puzzle configurations instead of states and disk moves instead of operators – and therefore belongs to the model's TOH level.

RH-Spatial

The RH-Spatial center corresponds to right IPS, an area that has been implicated in visual attention (e.g., Corbetta & Shulman, 2002). One function of this center is proposing direct operators based on comparisons between visuoperceptual inputs and internal visuospatial representations. Recall that direct operators increase the similarity between the current and ending states. The current state is typically a visuospatial representation in LH-Spatial and the ending state is typically visually available in the external environment. Another function of RH-Spatial is suppressing direct operators that are no longer relevant for future processing because they were (1) not selected for application, (2) selected but could not be applied because of an impasse, or (3) selected and successfully applied.

The exact nature of states varies from domain to domain, and therefore so does the sense in which they are similar or dissimilar to one another. Therefore, the RH-Spatial productions that propose direct operators belong to the TOH level of the model. One production compares the current and ending configurations and proposes ending moves, i.e., moves that directly transfer a disk that is out-of-place in the current configuration to its peg position in the ending configuration.

propose-ending-move
If *current-configuration* is the current state
ending-configuration is the ending state
disk is on peg *A* in *current-configuration*
disk is on a different peg *B* in *ending-configuration*
Then propose a *direct-move* of *disk* from peg *A* to peg *B*

This production is promiscuous in its proposal of direct moves. It does not verify that all of their preconditions are satisfied before proposing them. (Satisfying them is the concern of other centers.)

One design decision impacts RH-Spatial: whether top moves are also proposed. This is the *top-moves* decision. A top move transfers a disk from the top of one peg to the top of another. This is a visuospatially salient transformation, although it does not necessarily produce progress in problem solving. The *top-moves* decision is binary-valued. If the value is *yes*, then top moves are proposed in addition to the ending moves proposed by **propose-ending-move**. When this is the case, the following production is defined in LH-Spatial:

propose-top-move
If *current-configuration* is the current state
ending-configuration is the ending state
disk is on top of peg *A* in *current-configuration*
Then propose a *direct-move* of *disk* from peg *A* to the top of a different peg *B*

This production is domain-specific, depending as it does on the structure of puzzle configurations, and therefore belongs to the TOH level. Like **propose-ending-move**, it is also promiscuous in proposing moves whose preconditions are not necessarily satisfied. For example, it can propose moving a larger disk on top of a smaller disk, which is illegal according to the TOH task instructions. (Satisfying the preconditions of top moves is, once again, the province of other centers.) If the value of the *top-moves* design decision is *no*, then no additional productions are defined in RH-Spatial.

RH-Spatial also contains four productions that suppress direct operators that are no longer relevant. One suppresses direct operators that were not selected for application.

suppress-unselected-operator
If *selected-operator(directed-operator$_i$)*
directed-operator$_j$ $\neq$ *directed-operator$_i$*
Then suppress *direct-operator$_j$*

Another suppresses direct operators that were selected and successfully applied.

suppress-applied-selected-operator
If *selected-operator(directed-operator)*
current-state is the result of applying *direct-operator* to *previous-state*
Then suppress *direct-operator*

The third production suppresses direct operators that were selected but could not be applied because they were illegal.

suppress-illegal-selected-operator
If *selected-operator(directed-operator)*
direct-operator cannot be applied to *current-state* because it is illegal
Then suppress *direct-operator*

The final production suppresses direct operators that were selected but could not be applied.

suppress-unapplied-selected-operator
If *selected-operator(directed-operator)*
current-state is not the result of applying *direct-operator* to *previous-state*
Then suppress *direct-operator*

A selected operator cannot be applied if at least one of its preconditions is unsatisfied. This results in an impasse and the creation of a new goal to resolve this impasse – functions attributed to other centers. These productions are domain-general, and are therefore defined at the fronto-parietal level. They are important because RH-Spatial possesses a fixed supply of resources, and freeing them from no-longer-needed representations makes them available for future processing.

The Executive Centers

In Shallice's theory, executive function is implemented by the contention scheduler and SAS, which are localized to prefrontal areas. The contention scheduler selects between low-level schemas using conventional activation dynamics (i.e., excitation and inhibition). It is the seat of routine action. The SAS uses high-level schemas to organize the performance of complex and novel tasks. It does this indirectly, by biasing the activations of schemas, rather than directly, through the top-down exertion of control. It is the seat of non-routine action. The model of fronto-parietal interaction implements the contention scheduler and SAS using the computational mechanisms of Soar: preferences, goals, and (indirect) operators

LH-Executive

The LH-Executive center corresponds to left DLPFC. It is the contention scheduler of the model, to use Shallice's terminology, although it uses the computational mechanisms of Soar.

Graded, Unary Preferences. Prior versions of the model of fronto-parietal interaction implemented contention scheduling using the symbolic preferences and encapsulated decision procedure of Soar (Just & Varma, 2006; Newman et al., 2003). Specifically, when selecting between multiple operators, LH-Executive asserted preferences, each ordering a pair of operators based on their relative merit. This was too close to a re-implementation of Soar and too far from Shallice's proposal that contention scheduling is a graded, activation-based process. Moreover, it was not particularly robust because it lacked the complementary, though psychologically implausible, computational mechanisms of Soar required for reliable preference-based selection, such as its black-box decision procedure.

Therefore, a new LH-Executive has been implemented that is closer to Shallice's conception. Assume that multiple operators have been proposed by other centers. LH-Executive activates a preference for each operator. These are not the binary preferences of Soar. Rather, they are unary preferences, each representing the goodness of its associated operator. More precisely, it is the activation level of a unary preference that represents the goodness of its corresponding operator. Preferences accumulate activation over time through the iterative firing of heuristic productions. When the activation of a preference exceeds threshold, the associated operator is selected and the other preferences, which by definition have activations below threshold (otherwise they would have been selected first), are suppressed.

Given the isomorphism between operators and unary preferences, it is reasonable to ask whether the preferences are really necessary? Why not simply reason over the operators themselves? The short answer is that the activations of operators and preferences have different interpretations. (A longer answer is given in Chapter XI.) For example, direct operators are proposed by RH-Spatial. Therefore, the activation of a direct operator represents its goodness in the local sense of visuospatial attention. By contrast, preferences are activated by LH-Executive. Therefore, the activation of a preference represents its goodness in the local sense of selection, which is to say it represents the goodness of the associated operator in a global sense, relative to all other operators proposed by all other centers. Local goodness and global goodness are different notions, and therefore require different representations.

Design Decisions. Two design decisions impact LH-Executive. Unlike the other design decisions, they do not determine whether certain productions are included or excluded, but rather the style in which productions are written, specifically the schemes by which they direct activation. For this reason, their description is comparatively abstract.

The *accrual-scheme* decision governs how activation accrues to preferences over time. This decision is binary-valued. If the value is *additive*, on each cycle the activation of each preference is incremented by an amount proportional to the heuristic goodness of the associated operator. More precisely, if the goodness of *operator* is indicated by an amount of activation a, then the activation of the associated *preference* is incremented by a. If the threshold for selection is *THRESHOLD*, then it will take:

$$t = \frac{THRESHOLD}{a}$$

cycles for the activation of *preference* to exceed *THRESHOLD*, and for *operator* to be selected (assuming that another operator is not selected first). In other words, an *additive accrual-scheme* realizes a counter model of choice (Luce, 1986).

If the value of the *accrual-scheme* decision is *multiplicative*, then the activation of each preference is also incremented by an amount proportional to the heuristic goodness of the associated operator, but only on the first cycle. On each subsequent cycle, the activation is increased by a proportional amount. More precisely, if the proportion is *w*, then it will take :

$$t = \frac{\log_2(THRESHOLD) - \log_2(a)}{\log_2(1+w)}$$

cycles for the activation of *preference* to exceed *THRESHOLD*, and for *operator* to be selected (assuming once again that another operator is not selected first). Because *THRESHOLD* and *w* are constants, we can write this expression more concisely as:

$$t = B - C\log_2(a)$$

for suitably defined *B* and *C*. In other words, a *multiplicative accrual-scheme* realizes an information-theoretic model of choice (Luce, 1986).

The *preference-scheme* decision governs how the heuristic goodness of an operator is computed, and thus the amount of activation (denoted *a* above) directed to the associated preference. If the value of this decision is *absolute*, then it is the heuristic goodness of an operator alone that determines how much activation is directed to its associated operator. If the value of this decision is *relative*, then it is the heuristic goodness of an operator compared with competing operators that determines how much activation is directed to its associated operator. An *absolute* value for the *preference-scheme* is more compatible with Shallice's conception of contention scheduling; a relative value is more compatible with Soar's notion of binary preferences.

The *accrual-scheme* and *preference-scheme* decisions are orthogonal to one another. (In fact, this is true of all five design decisions.) Their values therefore combine factorally to yield four regimens for selecting among the available operators: *additive/absolute*, *additive/relative*, *multiplicative/absolute*, and *multiplicative/relative*. This deconstruction of selection into four regimens is an important achievement of the model. We will return to it in Chapter XI.

Heuristics. LH-Executive computes the goodness of an operator – specifically, the activation directed to the associated preference – using five heuristics, each implemented by a different production. Because the heuristics are domain-general, these productions belong to the fronto-parietal level.

The first heuristic is to prefer legal operators, or when the value of the *preference-scheme* decision is *relative*, to prefer legal operators over illegal operators. An operator is legal when all of its preconditions are satisfied; an illegal operator is one with one or more unsatisfied preconditions. The *absolute* form of this heuristic production is:

prefer-legal-absolute
If *direct-operator* is legal
Then activate *preference* by w_{legal}

The *relative* form is:

prefer-legal-relative
If *direct-operator*$_i$ is legal
direct-operator$_j$ ($i \neq j$) is illegal
Then activate *preference*$_i$ by w_{legal}

There are two things to notice. First, associated with each heuristic k is a weight w_k representing its informativeness relative to other heuristics. These weights are free parameters that are estimated when fitting particular data sets. In fact, two sets of weights will be estimated below, one for fitting data collected using a constrained presentation paradigm and the other for fitting data collected using an unconstrained presentation paradigm. The second thing to notice is that there is a systematic relation between the *absolute* and *relative* forms of heuristic productions. Therefore, only the *absolute* form of the remaining heuristic productions will be given. (The reader interested in the *relative* forms should consult the model's source code in Appendix A.)

The second heuristic is to prefer hill-climbing operators, or when the value of the *preference-scheme* decision is *relative*, to prefer hill-climbing operators over non-hill-climbing operators. A hill-climbing operator is one that increases the similarity between the current and ending states. In the TOH domain, for example, this is a move that transfers an out-of-place disk to its peg position in the ending configuration. The *absolute* form of this heuristic production is:

prefer-hill-climbing-absolute
If $\quad$ *direct-operator*$_i$ is a hill-climbing operator
Then $\quad$ activate *preference*$_i$ by $w_{hill\text{-}climbing}$

The third heuristic is a special case of the second one. It is to prefer the steepest-hill-climbing operator, or when the value of the *preference-scheme* decision is *relative*, to prefer steeper hill-climbing operators over shallower hill-climbing operators. The steepest hill-climbing operator is the one that most increases the similarity between the current and ending states. In the TOH domain, for example, this is the move that transfers the largest out-of-place disk to its peg position in the ending configuration. The *absolute* form of this heuristic production is:

prefer-steepest-hill-climbing-absolute
If $\quad$ *direct-operator*$_i$ is a hill-climbing operator
there exists no *direct-operator*$_j$ ($i \neq j$) that is a steeper hill-climbing operator than *direct-operator*$_i$
Then $\quad$ activate *preference*$_i$ by $w_{steepest\text{-}hill\text{-}climbing}$

The hill-climbing and steepest-hill-climbing heuristics are not redundant. Exploratory simulations revealed that the model produces too few errors if the former heuristic is omitted and too many errors if the latter heuristic is omitted.

The fourth heuristic is to prefer goal-based operators, or when the value of the *preference-scheme* decision is *relative*, to prefer goal-based operators over perception-based operators. Goal-based operators are indirect operators; they establish goals. (Perception-based operators are direct operators – they increase the similarity between the current and ending states.) They are important because they resolve the impasses that arise during the performance of complex and novels tasks. The *absolute* form of this heuristic production is:

prefer-goal-absolute
If   *indirect-operator*$_i$ is associated with *goal*$_i$
Then   activate *preference*$_i$ by w_{goal}

The fifth heuristic is a special case of the fourth heuristic. It is to prefer indirect operators associated with the topmost (i.e., most recent goal), or when the value of the *preference-scheme* decision is *relative*, to prefer indirect operators associated with the topmost goal over indirect operators associated with lower goals. (The height of a goal refers to its place in the goal stack.) The *absolute* form of this heuristic production is:

prefer-top-goal-absolute
If   *indirect-operator*$_i$ is associated with *goal*$_i$
there exists no *indirect-operator*$_j$ ($i \neq j$) that is associated with a *goal*$_j$ more recent than *goal*$_i$
Then   activate *preference*$_i$ by $w_{top\text{-}goal}$

The goal and top-goal heuristics are not redundant. Exploratory simulations revealed that the model produces too few errors if the former heuristic is omitted and too many errors if the latter heuristic is omitted.

Selection and Suppression. LH-Executive contains four additional productions. One recognizes when the activation of a preference exceeds threshold and selects the associated operator.

select-among-preferences
If the activation of *preference* exceeds *THRESHOLD*
operator is associated with *preference*
Then activate *selected-operator(operator)*

This production is domain-general, and therefore belongs to the fronto-parietal level.

Four productions recognize when an operator has been selected and suppress the partial products of the selection process. They are important because LH-Executive possesses a finite supply of computational resources, and freeing them from no-longer-needed representations makes them available for future processing. One production suppresses preferences once an operator has been selected.

suppress-preference
If *preference* exists
selected-operator(operator) exists
Then suppress *preference*

Three productions suppress the marker denoting which operator has been selected. If the selected operator's preconditions are satisfied, then it will be applied by RH-Spatial to the current state, producing a new current state. When this is the case, the existence of a new current state spurs suppression of the selected operator marker.

suppress-applied-preferred-operator-marker
If *selected-operator(operator)* exists
current-state is the result of applying *operator* to *previous-state*
Then suppress *selected-operator(operator)*

Another production suppresses the selected operator marker of the selected operator is illegal, and therefore could not be applied.

suppress-illegal-preferred-operator-marker
If *selected-operator(operator)* exists
operator cannot be applied to *current-state* because it is illegal
Then suppress *selected-operator(operator)*

Finally, if one or more of the selected operator's preconditions are unsatisfied, then an impasse will occur, and another center will establish a goal to resolve this impasse. When this is the case, it is the goal that spurs suppression of the selected operator marker.

suppress-unapplied-preferred-operator-marker
If *selected-operator(operator)* exists
operator is tied to *goal*
Then suppress *selected-operator(operator)*

Because the four suppression productions are domain-general, they belong to the fronto-parietal level.

RH-Executive

The RH-Executive center corresponds to right DLPFC. It is the SAS of the model, to use Shallice's terminology, although it is constructed from the computational mechanisms of Soar, not the localist connectionist technology that Shallice endorses (Cooper & Shallice, 2000). If the operator selected by LH-Executive cannot be applied to the current state because one or more of its preconditions are unsatisfied, then it is RH-Executive that activates a goal to resolve this impasse. It is also RH-Executive that proposes the indirect operators that achieve these goals.

The model implements the sophisticated perceptual strategy described in Chapter II. This strategy focuses on the largest out-of-place disk, clearing disks that block its movement to the peg position it occupies in the ending configuration. This strategy can impasse in two ways, which follow directly from the preconditions on moves. One precondition is that the disk being moved is the top disk on its peg. If this precondition unsatisfied, then the following production establishes a goal to unblock the disk by moving the disk directly above it.

propose-unblock-disk-goal
If *selected-operator(move)* exists
disk(move) is blocked from above
Then activate *unblock-disk-goal(disk(move))*

Another precondition is that the peg position to which the disk is being moved is unoccupied. If this precondition is unsatisfied, then the following production establishes a goal to unblock the peg position by moving the disk that occupies it.

propose-unblock-position-goal
If *selected-operator(move)* exists
destination-position(move) is occupied
Then activate *unblock-position-goal(destination-position(move))*

These two productions implement the goal-handling required by step (3) of the sophisticated perceptual strategy. They are domain-specific, and therefore belong to the model's TOH level.

One design decision impacts RH-Executive: whether goals are organized as a stack or an unordered collection. This is the *top-goal-moves-only* decision, and it is binary-valued. The two different organizations are contrasted in greater detail in Chapter XI. If the value of this decision is *yes*, then goals are organized as a stack. Specifically, only the topmost (i.e., most recent) goal spawns the proposal of indirect moves. When this is the case, two productions are defined in LH-Executive to implement the LIFO discipline of stacks. The first applies when the topmost goal is to unblock a disk by moving the disk directly above it to a *buffer-peg*.

propose-unblock-disk-move-top
If *unblock-disk-goal*$_i$ exists
disk is the immediately blocking disk
there exists no *goal*$_k$ ($i \neq k$) more recent than *unblock-disk-goal*$_i$
Then activate *indirect-move(disk*, *buffer-peg)*

The second production applies when the topmost goal is to unblock a peg position by moving the disk that occupies it to a *buffer-peg*.

propose-unblock-position-move-top
If *unblock-position-goal*$_i$ exists
disk is in *position(unblock-position-goal*$_i$*)*
there exists no *goal*$_k$ ($i \neq k$) more recent than *unblock-position-goal*$_i$
Then activate *indirect-move(disk*, *buffer-peg)*

The indirect moves proposed by these productions are sufficient to achieve the goals that arise when executing the sophisticated perceptual strategy. Needless to say, both belong to the TOH level of the model.

If the value of the *top-goal-moves-only* design decision is *no*, then the goals in RH-Executive form an unordered collection, i.e., do not follow the LIFO discipline of stacks. Specifically, all goals spawn the proposal of indirect moves. When this is the case, the following two productions are defined in RH-Executive. The first matches all goals to unblock a disk by moving the disk directly above it to a *buffer-peg*.

propose-unblock-disk-move-all
If *unblock-disk-goal*$_i$ exists
disk is the immediately blocking disk
Then activate *indirect-move(disk*, *buffer-peg)*

Notice that **propose-unblock-disk-move-all** is exactly like **propose-unblock-disk-move-top** except that it does not require *unblock-disk-goal*$_i$ to be topmost. The second production matches all goals to unblock a peg position by moving the disk that occupies it to a *buffer-peg*.

propose-unblock-position-move-all
If *unblock-position-goal*$_i$ exists
disk is in *position(unblock-position-goal*$_i$*)*
Then activate *indirect-move(disk*, *buffer-peg)*

Once again, **propose-unblock-position-move-all** is exactly like **propose-unblock-position-move-top** except that it does not require *unblock-position-goal*$_i$ to be topmost. These productions propose indirect moves sufficient to achieve the goals that arise when executing the sophisticated perceptual strategy, and for this reason they belong to the TOH level of the model.

Finally, RH-Executive contains a production that suppresses goals that have been established. This production is important because this center possesses a finite supply of computational resources, and suppressing no-longer-needed representations (i.e., goals that have been achieved) frees its resources for other processing.

suppress-satisfied-goal
If *goal* is satisfied in *current-state*
Then suppress *goal*

This production is domain-general, and therefore belongs to the fronto-parietal level of the model.

Collaborative Processing

The four centers collaborate to solve TOH problems, their pattern of communication a function of problem complexity and resource availability.

Simple problems can be solved largely through visuospatial processing, and therefore primarily recruit the Spatial centers. For example, the problem shown in Figure 20(a) requires only one move to solve. At the beginning of problem solving, the starting configuration is encoded in LH-Spatial as the current configuration. RH-Spatial proposes a number of direct operators. One is the ending move of transferring the lone out-of-place disk (i.e., the smallest one) from the peg it occupies in the current configuration (i.e., the left one) to the peg position occupies in the ending configuration (i.e., the right one). If the value of the *top-moves* decision is *true*, then top moves will also be proposed. (Recall that a top moves transfers a disk from the top of one peg to the top of another.) The two possible top moves are transferring the smallest disk from the left peg to the middle peg and transferring the medium disk from the right peg to the middle peg. Because there are no active goals, RH-Executive proposes no indirect operators.

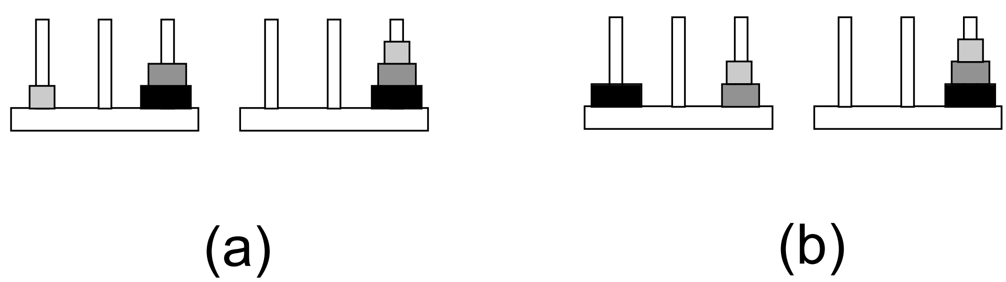


Figure 20. (a) A relatively easy problem. (b) A relatively difficult problem.

Next, LH-Executive activates a preference for each proposed move. If the value of the *preference-scheme* decision is *absolute*, then the heuristic productions will direct activation to preferences commensurate with the heuristic goodness of the associated operators. Preferences will continue to accrue activation according to the *accrual-scheme* in effect. When the activation of a preference exceeds threshold, the associated move will be selected.

Finally, the selected move will be applied to the current configuration by LH-Spatial (if possible), producing a new current configuration. If the selected move is the optimal move, then the new current configuration will be identical to the ending configuration, and the problem will be solved.

As this example illustrates, the Executive centers play a minimal role in the solution of simple problems. LH-Executive will be recruited to select between direct operators and RH-Executive will not be recruited at all. This suggests that the resource utilizations of the Executive centers should be relatively low when solving relatively simple problems (and when solving relatively simple portions of relatively difficult problems). It also suggests that simple problems can be solved even if the resources of the Executive centers are drastically reduced, as when simulating the effects of frontal lesions. The first of these predictions will be evaluated against neuroimaging data on normal young adults in Chapter X and the second against behavioral data on patients with frontal lesions in Chapter IX.

The Executive centers are increasingly recruited into the fronto-parietal network as difficulty increases, as when solving the problem shown in Figure 20(b). The direct moves proposed by RH-Spatial and selected by LH-Executive will not be applicable because not all of their preconditions will be satisfied. Impasses will occur, and RH-Executive will be recruited to resolve them. This center will establish goals keyed to the unsatisfied preconditions. It will also propose indirect moves that establish either the topmost goal or all active goals (depending on the value of the *top-goal-moves-only* decision). This processing will draw on its supply of computational resources. Of course, RH-Spatial will continue to propose direct moves based on comparisons between the current and ending configurations. LH-Executive will have to activate preferences for each of the direct moves proposed by RH-Spatial and each of the indirect moves proposed by RH-Executive – consuming its supply of computational resources.

One prediction, then, is that the resource utilization of the Executive centers should be relatively high when solving relatively difficult problems. Another is that if their resources are drastically reduced, as when simulating the effects of frontal lesions, the model should make many errors. These predictions will be evaluated against the neuroimaging data on normal young adults in Chapter X and the behavioral data on patients with frontal lesions in Chapter IX, respectively.

Chapter VIII

BEHAVIORAL MEASURES OF NORMAL YOUNG ADULTS

Chapter III reviewed recent behavioral studies of the TOH problem solving of normal young adults. These studies collect one temporal measure, time per individual move, and two error measures, number of moves and error rate. In this chapter, the 4CAPS model of TOH problem solving is evaluated against some of these data. One goal is to account for the variation in these data, both in the absolute sense of large correlations between model and human performance and in the relative sense of comparable or superior correlations to those achieved by competing models. To this end, all significance tests reported below were two-tailed and conducted at the α=.05 level. Another goal is to assess the design decisions that undergird the model. A final goal is to estimate values for certain of the model's parameters and to fix them as constants in subsequent data fitting.

The model contains nine free parameters, which can be divided into two sets. The first consists of the weights of the five heuristic productions in LH-Executive, described in Chapter VII. Two sets of values were estimated for these parameters, one for modeling optimal performance and the other for modeling errorful performance. The estimation process is described below. The second set of parameters consists of the resource capacities of the four model centers. These were fixed at 100.0 units each in the simulations reported in this chapter. This value is large enough that no center's resources were ever exhausted, not even when solving the most demanding problems considered in this dissertation. Tighter values for these parameters will be estimated in subsequent chapters, where the behavioral data on patient performance and the neuroimaging data on normal performance are addressed.

Individual Move Times

Two studies have collected individual move times from normal young adults. Ruiz (1987) had participants solve the standard 5-disk tower-to-tower problem. Anderson et al. (1993) had participants solve 4-disk problems isomorphic to (i.e., requiring the same number of moves and goal operations as) 4-disk tower-to-tower problems. They also had participants solve 5-disk problems isomorphic to 5-disk tower-to-tower problems. Ruiz (1987) employed a constrained presentation paradigm that kept participants on the optimal path. Anderson et al. (1993) employed an unconstrained presentation paradigm, but collected individual move times on those occasions when participants produced optimal solutions. These data were described more fully in Chapter III.

There are 48 variants of the model, one for each combination of the four binary-valued decisions and the one ternary-valued decision. Values for the first set of parameters – the weights of the five heuristic productions in LH-Executive – were selected such that each model variant could optimally solve the problems used by Ruiz (1987) and Anderson et al. (1993). These values are listed in the "Optimal Value" column of Table 11.

Table 11: Weights of the heuristic productions.

Heuristic Weight	Optimal Value	Errorful Distribution
w_{legal}	0.0125	U(0, 0.10)
$w_{hill\text{-}climbing}$	0.025	U(0, 0.15)
$w_{steepest\text{-}hill\text{-}climbing}$	0.05	U(0, 0.05)
w_{goal}	0.05	U(0, 0.25)
$w_{top\text{-}goal}$	0.10	U(0, 0.10)

The 48 model variants were run on the 4-disk problems of Anderson et al. (1993), the 5-disk problems of Anderson et al. (1993), and the 5-disk problems of Ruiz (1987). Table 12 lists the correlation of each model variant to each data set, as well as the average correlation to the 5-disk data sets.

Table 12: Correlations between individual move times and the 48 model variants.

Design Decisions					4-Disk		5-Disk	
preference-scheme	*accrual-scheme*	*top-moves*	*top-goal-moves-only*	*suppress-old-states*	Anderson et al. (1993)	Ruiz (1987)	Anderson et al. (1993)	Average
abs	mult	yes	yes	all	0.8477	0.8002	0.8897	0.851
abs	mult	yes	yes	non-goal	0.8477	0.8002	0.8897	0.851
abs	mult	yes	yes	none	0.8477	0.8002	0.8897	0.851
abs	mult	yes	no	all	0.8437	0.7969	0.8696	0.8369
abs	mult	yes	no	non-goal	0.8437	0.7969	0.8696	0.8369
abs	mult	yes	no	none	0.8437	0.7969	0.8696	0.8369
abs	mult	no	yes	all	0.8477	0.8002	0.8897	0.851
abs	mult	no	yes	non-goal	0.8477	0.8002	0.8897	0.851
abs	mult	no	yes	none	0.8477	0.8002	0.8897	0.851
abs	mult	no	no	all	0.8437	0.7969	0.8696	0.8369
abs	mult	no	no	non-goal	0.8437	0.7969	0.8696	0.8369
abs	mult	no	no	none	0.8437	0.7969	0.8696	0.8369
abs	add	yes	yes	all	0.8301	0.7832	0.8385	0.8127
abs	add	yes	yes	non-goal	0.8301	0.7832	0.8385	0.8127
abs	add	yes	yes	none	0.8301	0.7832	0.8385	0.8127
abs	add	yes	no	all	0.8245	0.7786	0.8253	0.8032
abs	add	yes	no	non-goal	0.8245	0.7786	0.8253	0.8032
abs	add	yes	no	none	0.8245	0.7786	0.8253	0.8032
abs	add	no	yes	all	0.8249	0.7828	0.8243	0.8045
abs	add	no	yes	non-goal	0.8249	0.7828	0.8243	0.8045
abs	add	no	yes	none	0.8249	0.7828	0.8243	0.8045
abs	add	no	no	all	0.8174	0.7773	0.81	0.7942
abs	add	no	no	non-goal	0.8174	0.7773	0.81	0.7942
abs	add	no	no	none	0.8174	0.7773	0.81	0.7942
rel	mult	yes	yes	all	0.8841	0.823	0.84	0.8317
rel	mult	yes	yes	non-goal	0.8841	0.823	0.84	0.8317
rel	mult	yes	yes	none	0.8841	0.823	0.84	0.8317
rel	mult	yes	no	all	0.8515	0.7806	0.7818	0.7812
rel	mult	yes	no	non-goal	0.8515	0.7806	0.7818	0.7812
rel	mult	yes	no	none	0.8515	0.7806	0.7818	0.7812
rel	mult	no	yes	all	0.7508	0.6859	0.8549	0.7845
rel	mult	no	yes	non-goal	0.7508	0.6859	0.8549	0.7845
rel	mult	no	yes	none	0.7508	0.6859	0.8549	0.7845
rel	mult	no	no	all	0.7219	0.6415	0.7865	0.7218
rel	mult	no	no	non-goal	0.7219	0.6415	0.7865	0.7218
rel	mult	no	no	none	0.7219	0.6415	0.7865	0.7218
rel	add	yes	yes	all	0.8649	0.7965	0.8389	0.8188
rel	add	yes	yes	non-goal	0.8649	0.7965	0.8389	0.8188
rel	add	yes	yes	none	0.8649	0.7965	0.8389	0.8188
rel	add	yes	no	all	0.8341	0.7826	0.786	0.7843
rel	add	yes	no	non-goal	0.8341	0.7826	0.786	0.7843
rel	add	yes	no	none	0.8341	0.7826	0.786	0.7843
rel	add	no	yes	all	0.5773	0.5239	0.7366	0.6425
rel	add	no	yes	non-goal	0.5773	0.5239	0.7366	0.6425
rel	add	no	yes	none	0.5773	0.5239	0.7366	0.6425
rel	add	no	no	all	0.5593	0.516	0.6751	0.6015
rel	add	no	no	non-goal	0.5593	0.516	0.6751	0.6015
rel	add	no	no	none	0.5593	0.516	0.6751	0.6015

Table 12 is given for completeness. More informative is Table 13, which averages the 4-disk correlations and average 5-disk correlations listed in Table 12 for each value of the five design decisions. These correlations provide some guidance on three of the decisions. For the *preference-scheme* decision, models that embody the *absolute* value correlate more highly with the 4-disk data than those that embody the *relative* value, although the difference is not reliable (z=0.43, p<0.67); the same is true of the 5-disk data (z=0.71, p<0.48). For the *accrual-scheme* decision, models that embody the *multiplicative* value correlate more highly with the 4-disk data than those that embody the *additive* value, although the difference is not reliable (z=0.29, p<0.78); the same is true of the 5-disk data (z=0.51, p<0.61). Finally, for the *top-moves* decision, models that embody the *yes* value correlate more highly with the 4-disk data than those that embody the *no* value, although the difference is not reliable (z=0.64, p<0.53); the same is true of the 5-disk data (z=0.50, p<0.62). The correlations listed in Table 13 provide no guidance on the two remaining design decisions. The 4-disk data do not discriminate between the possible values of the *top-goal-moves-only* decision (z=0.14, p<0.89); the same is true of the 5-disk data (z=0.20, p<0.85). Nor do the individual move time data discriminate between the possible values of the *suppress-old-states decision*: identical correlations are achieved to the 4-disk and 5-disk data regardless of which one is adopted.

Table 13: Correlation between individual move times of the human participants and the model averaged for each value of the five design decisions.

Design Decision	Value	4-Disk	Avg. 5-Disk
		r	
preference-scheme	*absolute*	0.84	0.83
	relative	0.78	0.76
accrual-scheme	*multiplicative*	0.83	0.82
	additive	0.79	0.77
top-moves	*yes*	0.85	0.82
	no	0.76	0.77
top-goal-moves-only	*yes*	0.82	0.81
	no	0.80	0.79
suppress-old-states	*all*	0.81	0.79
	non-goal	0.81	0.79
	none	0.81	0.79

This pattern of results can be interpreted in two ways: In an informal sense, the individual move time data offer guidance on three of the five design decisions. In other words, different correlations are achieved to the data depending on the values of these decisions – although the differences are not statistically reliable. In a formal sense, the individual move time data settle none of the five design decisions. The former interpretation will be adopted here. The reason is essentially pragmatic: the *N*s of the correlations listed in Tables 12 and 13 – 15 for the 4-disk data and 30 for the 5-disk data – are small enough that unrealistically large differences are required for statistical significance. For example, the model variant listed in the first row of Table 12 achieves a 0.85 correlation to the 4-disk data and an average 0.85 correlation to the 5-disk data. Although this fit appears to be quite good relative to the other model variants, it is statistically superior (at the .05 level) only to those with 0.42 (or lower) correlations to the 4-disk data and 0.62 (or lower) average correlations to the 5-disk data – of which there are 0 and 3, respectively.

Why is there so little variation in the correlations the different model variants achieve to the individual move time data? The answer, I believe, is *not* that the design decision are implementation details of no psychological consequence. Rather, it is that different values on these decisions are swamped by the large number of assumptions that the model variants share: those about the human cognitive neuroarchitecture (i.e., the model's bottom level, 4CAPS); those about the functional specializations and collaborative processing of frontal and parietal areas (i.e., the model's middle level); and those about the representations and solution strategy used (i.e., the model's top, domain-specific level). In other words, the differences between the model variants is a subtlety to which the statistical test (for the difference between correlations) is insensitive given the *N*s involved. For this reason, the average correlations listed in Table 13 will be interpreted in informally, at face value. The *preference-scheme*, *accrual-scheme*, and *top-moves* decisions will be considered settled and the *top-goal-moves-only* and *suppress-old-states* decisions will be considered open. The result is a reduction in the number of model variants that will be considered from 48 to 6.

Turning from the assessment of design decisions to the evaluation of the model's fit, the *preference-scheme*, *accrual-scheme*, and *top-moves* decisions were assigned their determined values of *absolute*, *multiplicative*, and *yes*, respectively, and the *top-goal-moves-only* and *suppress-old-states* decisions were arbitrarily assigned the values of *no* and *all*, respectively. (As Table 13 makes clear, the values of the latter two decisions have little impact on the model's fit to the individual move time data.) This model's individual move times on the 4-disk problems of Anderson et al. (1993) are plotted in Figure 21 alongside the human data. The correlation between the two is 0.85, indicating that the model does a good job (in an absolute sense) of accounting for the variation in the data ($z=3.06$, $p<0.003$).

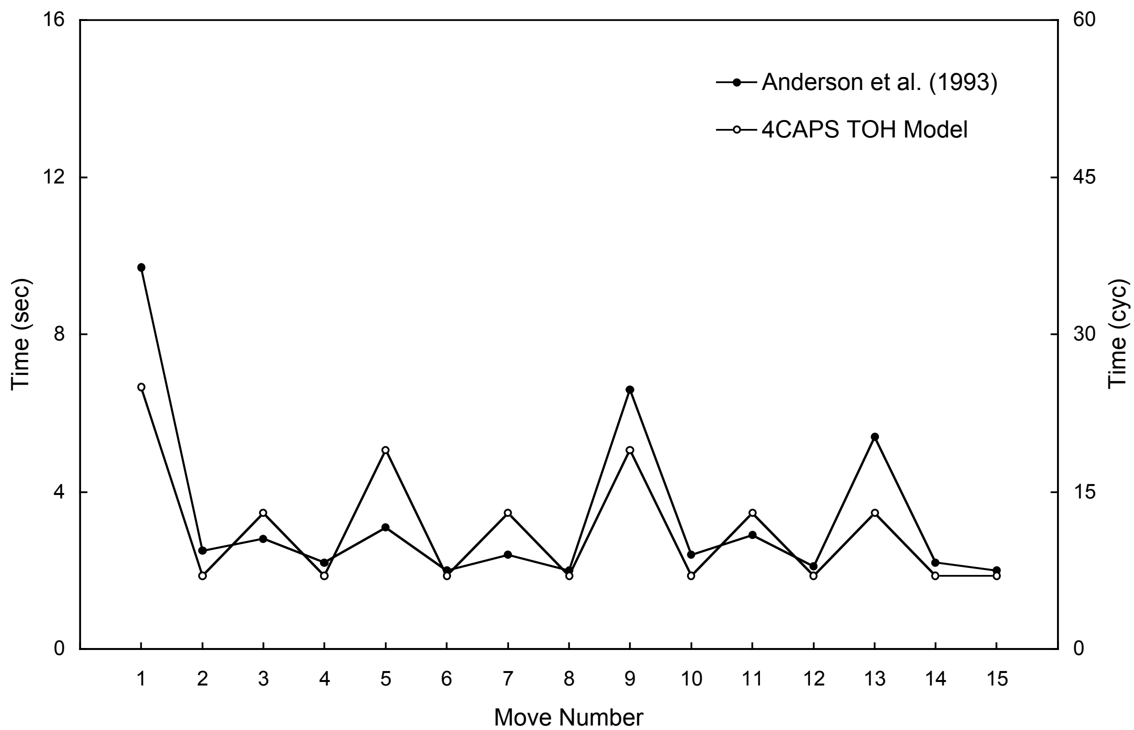


Figure 21. Individual move times for the 4-disk problems of the Anderson et al. (1993) participants and of the model.

The variation in the individual move times largely reflects the variation in the goal operations required by these moves. This is made clear by Table 14, which averages the times plotted in Figure 21 over moves that require generation of the same number of new goals. At this aggregate level, the correlation between the model and data is 0.94 (z=1.24, $p<0.22$).

Table 14: Average individual move times of the human participants and the model for the 4-disk problems of Anderson et al. (1993).

	Average Move Time	
New Goals	Anderson et al. (1993)	TOH Model
0	2.14 sec	7 cycles
1	3.38	13
2	4.85	19
3	9.7	25

This model variant was also run on the 5-disk problems of Ruiz (1987) and Anderson et al. (1993). The individual move times are plotted in Figure 22 alongside the data from each study. (These data can be directly compared because the studies used isomorphic problems.) The correlation between the model and the Ruiz (1987) data is 0.89 (z=5.32, $p<0.0001$) and the correlation between the model and the 5-disk Anderson et al. (1993) data is 0.80 (z=4.04, $p<0.0001$). Once again, the model does a good job (in an absolute sense) of accounting for the variation in the data.

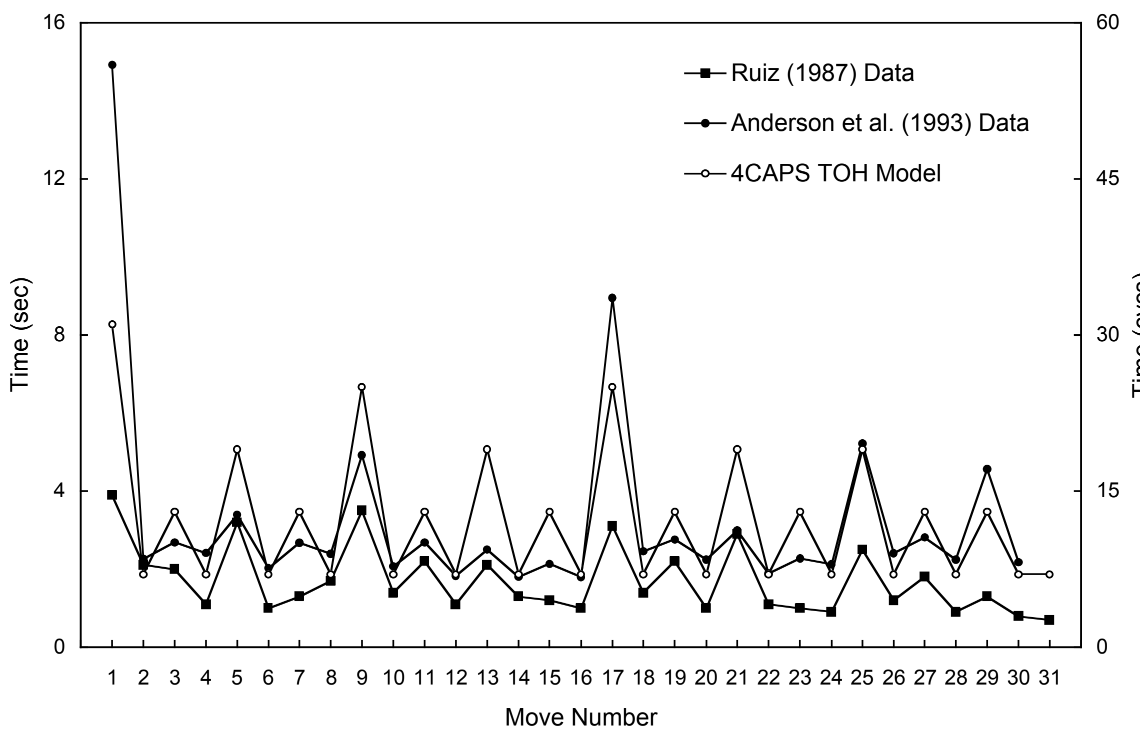


Figure 22. Individual move times for the 5-disk problems of the Ruiz (1987) and Anderson et al. (1993) participants and of the model.

This is made more evident by Table 15, which average the times plotted in Figure 22 over moves that require generation of the same number of new goals. At this aggregate level, the correlation between the model and the Ruiz (1987) data is 0.99 (z=2.65, p<0.008) and the correlation between the model and the 5-disk Anderson et al. (1993) data is 0.89 (z=1.42, p<0.38).

Table 15: Average individual move times of the human participants and the model for the 5-disk problems of Ruiz (1987) and Anderson et al. (1993).

	Average Move Time		
New Goals	Ruiz (1987)	Anderson et al. (1993)	TOH Model
0	1.17 sec	2.14 sec	7 cycles
1	1.63	2.82	13
2	2.68	3.53	19
3	3.3	6.94	25
4	3.9	14.92	31

The 4CAPS model fits the individual move times well in the absolute sense of variance accounted for. We can also ask how it compares to other models of the same data, of which there are two. The Anderson and Lebiere (1998) ACT-R model achieves a 0.89 correlation to the Ruiz (1987) data, which is of course indistinguishable from the 0.89 correlation achieved by the 4CAPS model. This ACT-R model achieves a 0.99 correlation to the 4-disk Anderson et al. (1993) data, which is reliably greater than the 0.80 correlation achieved by the 4CAPS model (z=3.79, p<0.0002). The Altmann and Trafton (2002) ACT-R model achieves a correlation of 0.99 to the 4-disk Anderson et al. (1993) data, which is reliably higher than the 0.85 correlation achieved by the 4CAPS model (z=3.41, p<0.0006). This ACT-R model achieves a 0.97 correlation to the 5-disk Anderson et al. (1993) data, which is also reliably higher than the 0.80 correlation achieved by the 4CAPS model (z=3.65, p<0.0004).

An easy conclusion to draw from this pattern of correlations is that the ACT-R models better account for the individual moves time data than the 4CAPS model. However, this neglects the critical fact that neither ACT-R model addresses all three data sets. Specifically, the Anderson and Lebiere (1998) model is silent on the 5-disk Anderson et al. (1993) data and the Altmann and Trafton (2002) model neglects the Ruiz (1987) data. Although generally speaking the 4CAPS model achieves a lower correlation to any single data set than the best-fitting ACT-R model, it is the only one that addresses all three data sets. In fact, it turns out that the 4CAPS model does quite well given the variability between the 5-disk data of Ruiz (1987) and Anderson et al. (1993). These studies had participants solve problems that were isomorphic to one another in the number of moves and goal operations required; sampled from the same population, undergraduates at elite private US universities; and instructed participants to use the goal recursion strategy. Given these commonalities, one would expect the correlation between the two data sets to be quite high. In fact, it is just 0.71. This is a useful reminder that even similar studies vary in a number of ways (e.g., the specific wording of instructions, the presentation software used), and these sources of variation accumulate to bound how well a single model can simultaneously account for *all* of the relevant data. The 0.71 correlation limits how well any *one* model can fit both 5-disk data sets. Relative to this standard, the 0.89 correlation the 4CAPS model achieves to the Ruiz (1987) data is quite good ($z=2.00$, $p<0.05$) and the 0.80 correlation it achieves to the 5-disk Anderson et al. (1993) data slightly less impressive ($z=0.78$, $p<0.44$). Moreover, the 0.97 correlation the Altmann and Trafton (2002) ACT-R model achieves to the 5-disk Anderson et al. (1993) data must be due in part to over-fitting; there is no way it can achieve a comparable correlation to the Ruiz (1987) data. A similar caveat applies to the Anderson and Lebiere (1998) model.

To summarize, the 4CAPS model of TOH problem solving provides a good account of the individual move time data, both in the absolute sense of variance accounted for and relative to competing models of the same data sets. Fitting these data provides guidance on three of the five design decisions, albeit in the informal/descriptive sense of "differences" rather than the formal/inferential sense of "reliable differences". Going forward, the value of the *preference-scheme* decision will be assumed to be *absolute*, the value of the *accrual-scheme* decision will be assumed to be *absolute*, and the value of the *top-moves* decision will be assumed to be *yes*. This reduces the space of model variants that will be considered from 48 to 6, or said another way, reduces the number of free parameters available for fitting data by three. Two design decisions remain open, one governing the organization of goals in RH-Executive (*top-goal-moves-only*) and the other the suppression of old states in LH-Spatial (*suppress-old-states*). A final achievement was the identification of heuristic weights under which all model variants can optimally solve the relatively difficult 4-disk and 5-disk problems employed by Anderson et al. (1993) and Ruiz (1987). These weights will be fixed in all subsequent modeling of optimal TOH problem solving. This represents a further reduction in degrees of freedom going forward.

Number of Moves

Another behavioral measure that has been collected from normal young adults is the number of moves required to solve TOH problems. This is an error measure in that it reflects deviations from optimality (i.e., minimum-length solutions). Anderson et al. (1993) collected this measure for a set of eight 4-disk problems and a set of eight 5-disk problems. Each set contained two instances from each of the tower-to-tower, tower-to-spread, spread-to-tower, and spread-to-spread classes. Despite their different surface appearances, all of the 4-disk problems were isomorphic to 4-disk tower-to-tower problems and all of the 5-disk problems were isomorphic to 5-disk tower-to-tower problems. These data, which were described more fully in Chapter III, constitute a test of the model's ability to account for errorful problem solving.

To account for the Anderson et al. (1993) error data, a second set of weights for the five heuristic productions of LH-Executive was chosen. Recall that the first set, listed above in the "Optimal Value" column of Table 11, was selected for simulating optimal performance. The second set, listed in the "Errorful Distribution" column of the same table, was selected for simulating errorful performance. The errors people make during problem solving are in part unsystematic, and therefore accounting for the error data requires that the model make optimal moves on some occasions and suboptimal moves on other occasions. However, the 4CAPS cognitive architecture is deterministic, and therefore its models do not make unsystematic errors. For this reason, one goal of this dissertation project was to introduce stochasticity. This was accomplished by modifying the model's middle level, which captures the interaction between frontal and parietal areas. Specifically, the weights of LH-Executive's heuristic productions were changed from the scalars used to simulate optimal performance to random variables with uniform distributions. These stochastic weights will be used in all subsequent modeling of errorful problem solving.

Six model variants were considered, one for each combination of the possible values of the *top-goal-move-only* and *suppress-old-states* decisions. First, consider the 4-disk data of Anderson et al. (1993). Each model variant solved each 4-disk problem 20 times. The average number of moves that each variant required to solve each problem is listed in Table 16, as are the human data. No variant achieves a correlation with the data that differs reliably from 0. (The highest positive correlation (0.21) is attained when the value of the *top-goal-moves-only* decision is *no* and the value of the *suppress-old-states* decision is *non-goal* (z=0.34, $p<0.74$).)

Table 16: Number of moves required by human participants and the six model variants for the 4-disk problems of Anderson et al. (1993).

		top-goal-moves-only = yes			*top-goal-moves-only = no*		
Problem	Data	*all*	*non-goal*	*none*	*all*	*non-goal*	*none*
Tower-to-Tower 1	21.4	18.2	18.6	18.2	17.2	19.0	18.7
Tower-to-Tower 2	16.9	18.1	17.7	19.8	18.2	17.5	18.7
Tower-to-Spread 1	19.3	20.9	21.2	21.8	19.4	18.2	19.8
Tower-to-Spread 2	21.1	18.1	18.8	17.6	18.1	18.8	17.3
Spread-to-Tower 1	21.2	18.4	18.7	17.0	18.2	18.9	18.0
Spread-to-Tower 2	17.0	17.4	18.3	17.9	17.6	17.8	16.6
Spread-to-Spread 1	18.5	20.0	22.0	20.2	19.6	18.6	17.8
Spread-to-Spread 2	18.4	20.8	20.8	23.2	22.5	21.7	22.2
Average	19.2	19.0	19.5	19.5	18.9	18.8	18.6
r		-0.06	-0.04	-0.41	-0.26	0.21	-0.08

Next, consider the 5-disk data of Anderson et al. (1993). Once again, each model variant solved each 5-disk problem 20 times. The average number of moves that each variant required to solve each problem is listed in Table 17. Once again, no variant achieves a correlation with the data that differs reliably from 0. (The highest positive correlation (0.51) is attained when the value of the *top-goal-moves-only* decision is *no* and the value of the *suppress-old-states* decision is *none* (z=0.89, $p<0.38$).)

Table 17: Number of moves required by human participants and the six model variants for the 5-disk problems of Anderson et al. (1993).

		top-goal-moves-only = yes			*top-goal-moves-only = no*		
Problem	Data	*all*	*non-goal*	*none*	*all*	*non-goal*	*none*
Tower-to-Tower 1	55.4	41.4	44.2	46.5	45.6	41.8	53.6
Tower-to-Tower 2	54.2	43.9	44.6	43.8	49.8	53.3	50.6
Tower-to-Spread 1	53.2	52.1	57.7	51.6	52.7	52.5	50.4
Tower-to-Spread 2	49.0	56.3	58.8	55.8	48.5	54.7	54.5
Spread-to-Tower 1	58.2	44.2	44.6	42.8	48.5	45.6	50.2
Spread-to-Tower 2	50.6	44.1	44.7	42.7	41.4	49.8	43.7
Spread-to-Spread 1	58.5	59.8	63.7	63.4	55.5	48.9	57.7
Spread-to-Spread 2	48.6	57.4	68.3	69.2	52.15	48.7	46.9
Average	53.5	49.9	53.3	52.0	49.3	49.4	50.1
r		-0.24	-0.32	-0.28	0.27	-0.48	0.51

To summarize the story thus far, the model shows little ability to account for the number of moves data of Anderson et al. (1993). This failure holds regardless of the values of the *top-goal-moves-only* and *suppress-old-states* decisions. This can be seen more clearly in Table 18, which averages the correlations in Tables 16 and 17 separately for each possible value of these decisions. No correlation differs reliably from 0. In other words, the number of moves data of Anderson et al. (1993) settle neither of the remaining design decisions.

Table 18: Correlation between the number of moves required by human participants and the model averaged for each value of the *top-goal-moves-only* and *suppress-old-states* decisions.

Design Decision	Value	r
top-goal-moves-only	*yes*	-0.23
	no	0.03
suppress-old-states	*all*	-0.07
	non-goal	-0.17
	none	-0.06

How dire is the failure of the 4CAPS model to account for the number of moves data of Anderson et al. (1993)? It is notable that although several models address the temporal data of this study, only one addresses its error data. Specifically, Altmann and Trafton (2002) computed the *average* number of moves their ACT-R model requires to solve the eight 4-disk problems and the *average* number of moves it requires to solve the eight 5-disk problems; these values are 17.7 and 50.6, respectively. They compare favorably with the average number of moves required by the Anderson et al. (1993) participants: 19.2 for the 4-disk problems and 53.5 for the 5-disk problems. The 4CAPS model can also be evaluated against the averaged data. Table 19 lists, for each model variant, the average number of moves required to solve the eight 4-disk problems and the eight 5-disk problems; it also lists the averaged human data. Informally speaking, each variant of the 4CAPS model accounts for the aggregate data about as well as the Altmann and Trafton (2002) ACT-R model. However, the six model variants are themselves indistinguishable. For this reason, the average number of moves data of Anderson et al. (1993) offer no guidance on the two remaining design decisions.

Table 19: Average number of moves required by human participants and the six model variants to solve the 4-Disk and 5-Disk problems of Anderson et al. (1993).

		top-goal-moves-only = yes			*top-goal-moves-only = no*		
Problem	Data	*all*	*non-goal*	*none*	*all*	*non-goal*	*none*
4-Disk	19.2	19.0	19.5	19.4	18.8	18.8	18.6
5-Disk	53.5	49.9	53.3	51.9	49.3	49.4	50.9

Summary

This chapter has evaluated the 4CAPS model against the behavioral data on the TOH problem solving of normal young adults. It fits these data well, both in the absolute sense of high correlations and relative to the correlations achieved by competing models. This process also provided guidance on three of the five design decisions.

The model provides a good account of the individual move time data of Anderson et al. (1993) and Ruiz (1987). In fact, it is comparable to the combined account offered by two existing ACT-R models (modulo the inherent variability of the 5-disk data sets). Fitting the temporal data required selecting weights for the heuristic productions of LH-Executive. These weights will be fixed for all subsequent modeling of optimal performance. The individual move time data offered guidance on three design decisions: the *preference-scheme* of LH-Executive is *absolute*, the *accrual-scheme* of LH-Executive is *multiplicative*, and the *top-moves* decision of RH-Spatial is *yes*.

The model does a poor job accounting for the number of moves data of Anderson et al. (1993) on a problem-by-problem basis. However, when this measure is averaged over problems with the same number of disks, the correspondence between the 4CAPS model and the data improves dramatically. This is true in an absolute sense and relative to the account offered by the ACT-R model of Altmann and Trafton (2002). Fitting the error data required selecting weights for the heuristic productions of LH-Executive. These weights are not scalars, but rather uniformly-distributed random variables. They will be fixed for all subsequent modeling of errorful performance. Unlike the individual move time data, the (average) number of moves data offered no guidance on the remaining design decisions. Therefore, the *top-goal-moves-only* and *suppress-old-states* decisions will continue to be assessed in subsequent chapters.

CHAPTER IX

BEHAVIORAL MEASURES OF LESION PATIENTS

This chapter evaluates the model against the behavioral data on the TOH problem solving of patients with frontal lesions. The model's ability in the regard is rooted in its architectural level. 4CAPS construes cortical information processing as a collaboration among multiple centers, each with functional specializations relevant for the task at hand. Computation within a center – the storage and processing of representations – is fueled by a fixed supply of resources. A lesion to a brain area is simulated by drastically reducing the resources available in the corresponding center. This limits the center's ability to perform the cognitive functions for which it is specialized, which in turn limits task performance. This technique has enabled a 4CAPS model of sentence comprehension to account for the cognitive deficits of a patient with a lesion to Broca's area (Just & Varma, 2006).

This technique can be applied to TOH problem solving. One goal of this chapter is to evaluate the model against the behavioral data on the TOH problem solving of patients with frontal lesions. Another goal is to assess the *top-goal-moves-only* design decision, which governs the organization of goals in RH-Executive. (The other open design decision, *suppress-old-states*, governs the maintenance of visuospatial representations in LH-Spatial. Because there is no usable data in the literature on the TOH problem solving of patients with lesions to corresponding left parietal cortex, no attempt was made to assess this design decision in this chapter. It will be taken up again in Chapter X.) Addressing these data requires estimating for the first time the second set of model parameters, the resource capacities of the four centers. (Recall that the first set of model parameters, the weights of the heuristic productions in LH-Executive, was estimated twice in Chapter VIII, once for modeling optimal performance and once for modeling errorful performance.)

This chapter is organized as follows. First, the Morris et al. (1997a) study is reviewed and the reason why its data were not modeled is given. Second, the Morris et al. (1997b) study is reviewed and the reason why its data were not modeled is also given. Third, the model is evaluated against the Goel and Grafman (1995) data in the form reported by Goel et al. (2001). These data consist of three measures: proportion of problems solved in the allotted time, number of moves, and overall solution time. These measures have been collected from four populations: normal young adults, patients with left frontal lesions, patients with right frontal lesions, and patients with bilateral frontal lesions. The Goel et al. (2001) data are exceedingly rich, and therefore provide a good benchmark against which to evaluate the model's ability to account for behavioral effects of brain lesions.

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997a)

Morris et al. (1997a) compared the performance of intact normals, patients with left frontal lesions, and patients with right frontal lesions when solving TOH problems. A second factor was whether problems did or did not induce goal-subgoal conflict (Goel & Grafman, 1995). Goal-subgoal conflict occurs when the first move on the optimal solution sequence is perceptually counter-intuitive, i.e., decreases rather than increases the similarity between current (staring) configuration and the ending configuration. Making this move requires the use of executive function, which is commonly attributed to frontal areas.

The model is capable in-principle of accounting for the Morris et al. (1997a) data. It experiences heightened contention between routine and non-routine action, to use Shallice's (1982) terminology, on problems that induce goal-subgoal conflict. RH-Spatial proposes direct moves that increase the similarity between the current and ending configurations. RH-Executive, by contrast, is not a slave to similarity. Rather, it proposes indirect moves that achieve goals. When faced with a problem that induces goal-subgoal conflict, it is RH-Executive that will propose the optimal first move. LH-Executive is specialized for choosing between the moves proposed by other centers, including RH-Spatial and RH-Executive. It does this through preferential reasoning – the activation of preferences by heuristic productions. The model predicts that lesions to left or right DLPFC will produce suboptimal performance on problems that induce goal-subgoal conflict. A lesion to right DLPFC, operationalized as a drastic reduction in the resource supply of RH-Executive, will impair the proposal of indirect moves, including the optimal first move. A lesion to left DLPFC, operationalized as a drastic reduction in the resource supply of LH-Executive, will degrade preferential reasoning, decreasing the likelihood that the optimal first move proposed by RH-Executive will be selected instead of one of the suboptimal moves proposed by RH-Spatial.

Unfortunately, the model could not be evaluated against the Morris et al. (1997a) data because not all of the stimuli shown in the published report meet the criteria listed in the Method section. The study employs the four problem types that result from the orthogonal combination of two levels of optimal solution length (4 vs. 5 moves) and two levels of goal-subgoal conflict (no-conflict vs. conflict). Examples of each problem type are given in the published report. However, the example 5-move/no-conflict problem is in fact a 5-move/conflict problem. This is because the optimal first move does not transfer a disk to its peg position in the ending configuration, but rather to a buffer peg position. In addition, the 4-move/conflict problem is in fact a 4-move/no-conflict problem: the first move does not transfer a disk to a peg position different than the one it occupies in the ending configuration, but rather directly to it. Because problem stimuli are missing for two of the study's four cells, the model could not be applied to the Morris et al. (1997a) data.

Morris, Miotto, Feigenbaum, Bullock, and Polkey (1997b)

Morris et al. (1997b) compared the performance of intact normals, patients with left frontal lesions, and patients with right frontal lesions on TOH problems where the optimal solution sequence and the next-best solution sequence were either similar or dissimilar in length. There was a main effect of group, with right frontal patients making more moves than left frontal patients and intact normals. However, there was no main effect of solution length and there was no interaction between these variables.

Unfortunately, the model could not be evaluated against the Morris et al. (1997b) data because one problem stimulus in the published report does not meet the design criteria given in the Method section. The study employs four problem types, the result of orthogonally varying two levels of optimal solution length (6 vs. 7 moves) and two levels of the similarity between the length of the optimal and next-best solution sequences (similar vs. dissimilar). For similar problems, the next-best solution sequence was 1 move longer than the optimal solution sequence; for dissimilar problems, the difference was 3 moves. An example of each problem type is given in the published report. However, there is an issue with the 7-move/dissimilar problem: its optimal solution sequence is actually 6 moves in length. It is possible that the problem was depicted incorrectly in a trivial way: it differs by the placement of a single disk from a problem that appears in another figure in the published report, one illustrating the experimental software. Although this problem requires 7 moves to solve, its next-best solution is only 2 moves longer than the optimal one, not the 3 moves required by the design criteria, so this is not the 7-move/dissimilar problem either. Because no problem stimulus is available for one of the study's four cells, the model could not be applied to the Morris et al. (1997b) data.

Goel, Grafman, and Pullara (2001)

Goel and Grafman (1995) investigated the TOH problem solving of patients with frontal lesions. Unlike the Morris et al. (1997a; 1997b) studies, there is no uncertainty regarding the problems that participants solved: all nine appear in the published report and match the design criteria listed in the Method section.

Goel and Grafman (1995) tested three groups of patients – those with left frontal lesions, those with right frontal lesions, and those with bilateral frontal lesions – and a control group of intact normals. Three behavioral measures were collected: proportion of times a problem was solved in less than two minutes, number of moves, and overall solution time. These measures were combined into a composite score that was plotted separately for each group and for each problem. This presentation of the data has the advantage of supporting inferences about the differential specializations of left and right frontal areas but the disadvantage of conflating the diagnostic information provided by the each component measure. Because Goel and Grafman (1995) found few differences between the three patient groups, Goel et al. (2001) combined them into a single frontal lesion group. This loss of resolution at the group (i.e., lesion site) level was offset by the decomposition of the score measure back into its three component measures. The 4CAPS model was evaluated against the Goel et al. (2001) presentation of the data, for two reasons. First, this is the presentation against which they evaluated their 3CAPS model of TOH problem solving, and it therefore facilitates comparisons between the two models. Second, whereas the numerical values of the data as plotted in Goel et al. (2001) are still available, they are no longer available for the Goel and Grafman (1995) presentation (Goel, personal communication).

Fitting the 4CAPS model to these data requires estimating a number of parameters. One is the mapping between real time and model time. This is necessary for accounting for one of the measures, the proportion of problems solved in the allotted time. In preliminary simulation work, the model solved the most difficult of the nine problems (i.e., the one with the longest optimal solution length) 20 times under time allotments stepped in increments of 10 cycles. The maximum correlation between the model and the intact normals across all three measures is achieved when the time allotment is 290 cycles, which corresponds to a temporal mapping of 290 cycles/120 second=2.42 cycles/second.

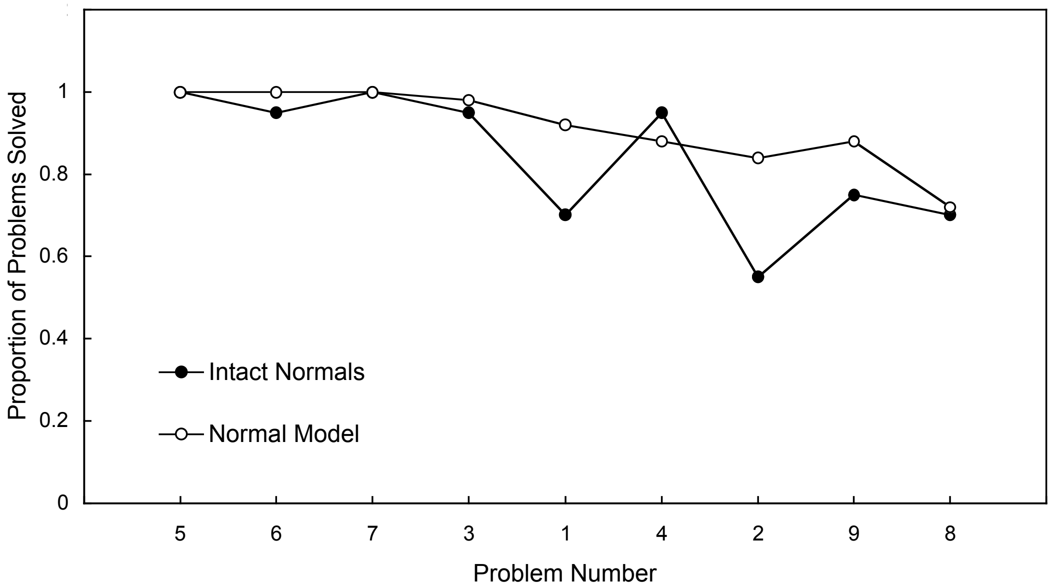
The remaining parameters are the resource capacities of the LH-Executive and RH-Executive centers when simulating the four groups: normal controls, left frontal patients, right frontal patients, and bilateral frontal patients. (Because the LH-Spatial and RH-Spatial centers are not the focus of this chapter, they were assigned the same default resource capacities used in Chapter IX.) Therefore, there are potentially (2 centers x 4 groups =) 8 free parameters. To minimize the degrees of freedom, these were reduced to 4 free parameters: the resource capacities of LH-Executive and RH-Executive when simulating intact normal controls; the resource capacity of LH-Executive when simulating patients with left frontal and bilateral frontal lesions; and the resource capacity of RH-Executive when simulating patients with right frontal or bilateral frontal lesions. Each parameter was estimated twice, once for each value of the *top-goal-moves-only* decision, using the following four-step process: First, "minimally perfect" resource capacities for the LH-Executive and RH-Executive centers were estimated by having the model solve the most difficult problem 20 times and recording the peak resource consumption in each center. Second, a grid search was performed, decreasing the minimally perfect resources of LH-Executive and RH-Executive by 0.5 units on each step, to estimate the values that maximized the correlation between the model and the intact normals across all three measures. These values, the resource capacities of the Executive centers of the intact normal model, are listed in Table 20 for both values of the *top-goal-moves-only* decision. Third, the reduced resource capacity of LH-Executive corresponding to a left frontal or bilateral frontal lesion was estimated as the minimum value under which the model could solve the most difficult problem at least once in 20 attempts. Fourth, the reduced resource capacity of RH-Executive corresponding to a right frontal or bilateral frontal lesion was estimated as the minimum value under which the model could solve the most difficult problem at least once in 20 attempts. The resource capacities of the left frontal, right frontal, and bilateral frontal models are listed in Table 20, again for both values of the *top-goal-moves-only* decision.

Table 20: Resource capacities of the model centers used to simulate Goel et al. (2001).

		Frontal Lesions		
Center	Intact Normals	Left	Right	Bilateral
top-goal-moves-only=yes				
LH-Executive	9.0	6.0	9.0	6.0
RH-Executive	5.0	5.0	2.5	2.5
LH-Spatial	100.0	100.0	100.0	100.0
RH-Spatial	100.0	100.0	100.0	100.0
top-goal-moves-only=no				
LH-Executive	9.0	6.0	9.0	6.0
RH-Executive	4.5	4.5	3.5	3.5
LH-Spatial	100.0	100.0	100.0	100.0
RH-Spatial	100.0	100.0	100.0	100.0

The TOH model was run with the resource capacities listed in Table 20 to simulate the performance of intact normals, left frontal patients, right frontal patients, and bilateral frontal patients. Specifically, each model solved each of the nine problems 50 times. Three measures were recorded: proportion of problems solved in the allotted time (290 cycles), number of moves, and overall solution time. Recall that Goel et al. (2001) collapsed the three patient groups into an aggregate frontal group. Therefore, the performance of the left frontal, right frontal, and bilateral frontal models was averaged, each weighted by the size of the corresponding patient group (6, 8, and 6, respectively), to yield the performance of the aggregate frontal model. This process was repeated for both values of the *top-goal-moves-only* decision.

We are now in a position to evaluate the fit of the 4CAPS model to the Goel et al. (2001) data and to compare this to the fit achieved by their 3CAPS model. The first measure to be considered is the proportion of problems solved in the allotted time. The data for the intact normals are plotted in the top panel of Figure 23 and the data for the frontal patients in the bottom panel. There is a main effect of group, with intact normals solving a greater proportion of problems than frontal patients. For both groups, there is an inverse relation between problem difficulty and proportion of problems solved: the more difficult a problem, the smaller the proportion of times it was solved. The performance of the normal and lesioned models is also plotted in the top and bottom panels, respectively, of Figure 23. Qualitatively speaking, the models reproduce the main effect of group: the normal model solves a greater proportion of problems than the lesioned model. They also display the observed inverse relation between problem difficulty and proportion of problems solved.



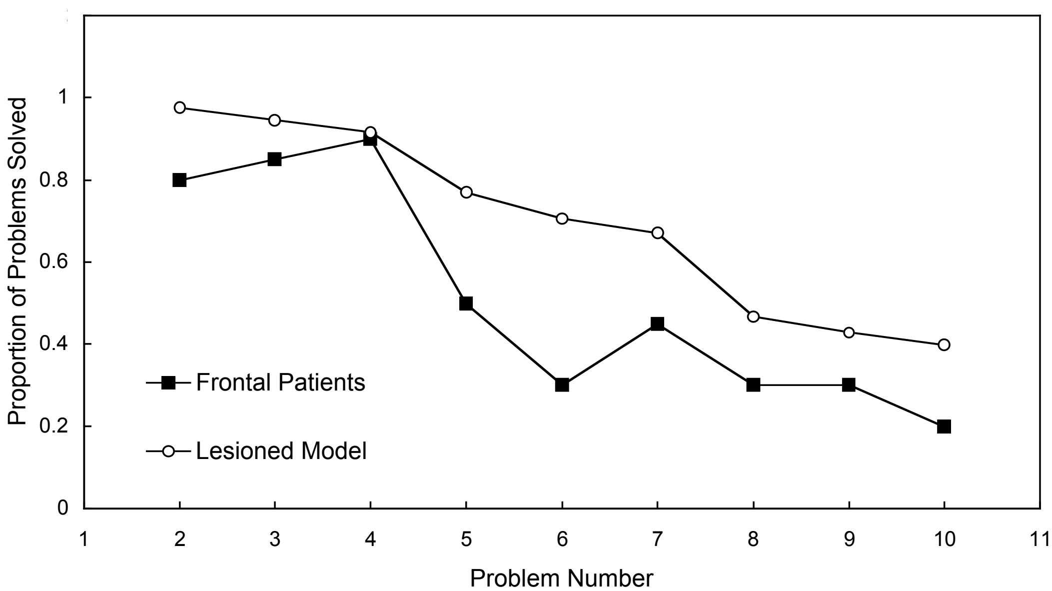


Figure 23. Proportion of problems solved in the allotted time (two minutes) (a) by the intact normals of Goel et al. (2001) and the normal model and (b) by the frontal patients and the lesioned model.

The correlation between the performance of the intact normals and frontal patients and that of the normal and lesioned 4CAPS models is listed in Table 21 for both values of the *top-goal-moves-only* decision. A higher correlation results when the value of the *top-goal-moves-only* decision is *no* rather than *yes*, suggesting that the goals in RH-Executive are organized like a set, not a stack. However, this difference is not reliable (z=0.59, p<0.56). Table 21 also lists the correlation between the performance of the intact normals and frontal patients and that of the normal and lesioned 3CAPS models of Goel et al. (2001). The 4CAPS models where the value of the *top-goal-moves-only* decision is *no* achieve a correlation to the data comparable to that of the 3CAPS models (z=0.10. p<0.92). The 4CAPS models where the value of the *top-goal-moves-only* decision is *yes* achieve a lower correlation to the data than that of the 3CAPS models, but the difference is not reliable (z=-0.49, p<0.63).

Table 21: Correlation between the proportion of the Goel et al. (2001) problems solved by normal and lesioned patients and models for each value of the *top-goal-moves-only* decision.

	r
4CAPS	
top-goal-moves-only=yes	0.83
top-goal-moves-only=no	0.91
3CAPS	0.90

The second measure to be considered is an error measure, number of moves. Recall that in Chapter VIII, the model demonstrated a modest ability to account for this measure as collected from normal young adults. The data for the intact normals are plotted in the top panel of Figure 24 and the data for the frontal patients are plotted in the bottom panel. Although there is no difference between the groups, there is a direct relation between problem difficulty and number of moves, with more difficult problems requiring more moves to solve. The number of moves required by the normal and lesioned models are also plotted in the top and bottom panels, respectively, of Figure 24. Informally speaking, the models provide a good account of the data, exhibiting no effect of group but a positive relation between problem difficulty and number of moves.

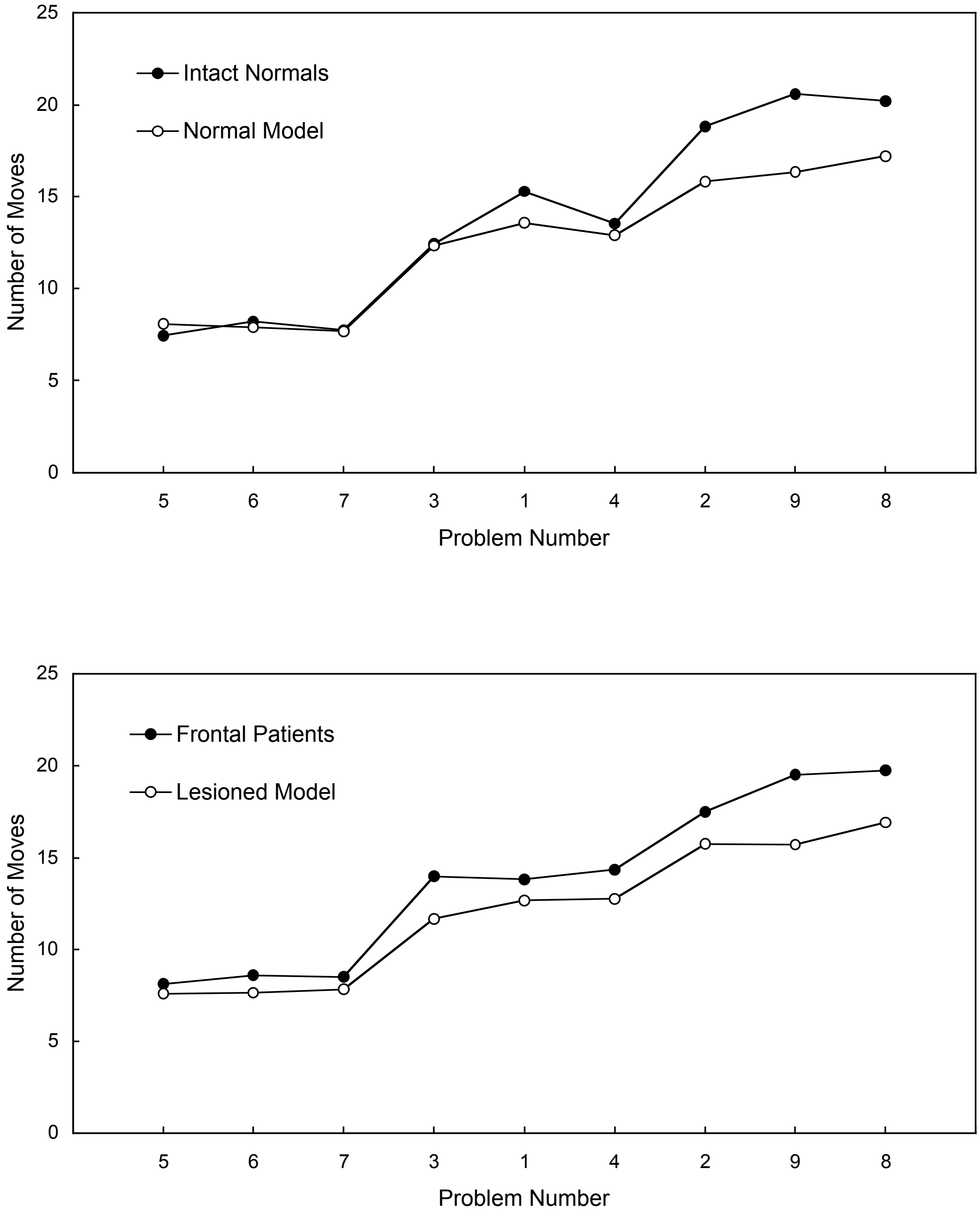


Figure 24. Number of moves required (a) by the intact normals of Goel et al. (2001) and the normal model and (b) by the frontal patients and the lesioned model.

These informal observations can be made more precise. The correlation between the performance of the intact normals and frontal patients and that of the normal and lesioned 4CAPS models is listed in Table 22 for both values of the *top-goal-moves-only* decision. A higher correlation results when the value of the *top-goal-moves-only* decision is *no* rather than *yes*. However, this difference is not reliable (z=0.60, p<0.55). Table 22 also lists the correlation between the performance of the intact normals and frontal patients and that of the normal and lesioned 3CAPS models of Goel et al. (2001). The 4CAPS models where the value of the *top-goal-moves-only* decision is *no* achieve a higher correlation to the data than the 3CAPS models, but this advantage is not reliable (z=1.41, p<0.16). The 4CAPS models where the value of the *top-goal-moves-only* decision is *yes* also achieve a higher correlation to the data that the 3CAPS models, but this advantage is also not reliable (z=0.81, p<0.42).

Table 22: Correlation between the number of moves required to solve the Goel et al. (2001) problems by the intact normal and frontal lesion patients and the normal and lesioned models for each value of the *top-goal-moves-only* decision.

	r
4CAPS	
top-goal-moves-only=yes	0.98
top-goal-moves-only=no	0.99
3CAPS	0.95

The final measure to be considered is a temporal one, overall solution time. Recall that in Chapter VIII, the 4CAPS model accounted for the individual move times of normal young adults. The challenge here is to generalize this success to a different set of problems and to a different population, patients with frontal lesions. The data for the intact normals are plotted in the top panel of Figure 25 and the data for the frontal patients in the bottom panel. There is an effect of group: frontal patients require more time than intact normals. There is also a direct relation between problem difficulty and overall solution time, with more difficult problems requiring more time to solve. The overall solution times for the normal and lesioned models are also plotted in the top and bottom panels, respectively, of Figure 25. Qualitatively speaking, the model does a good job accounting for the data, displaying both a group difference and a direct relation between problem difficulty and overall solution time.

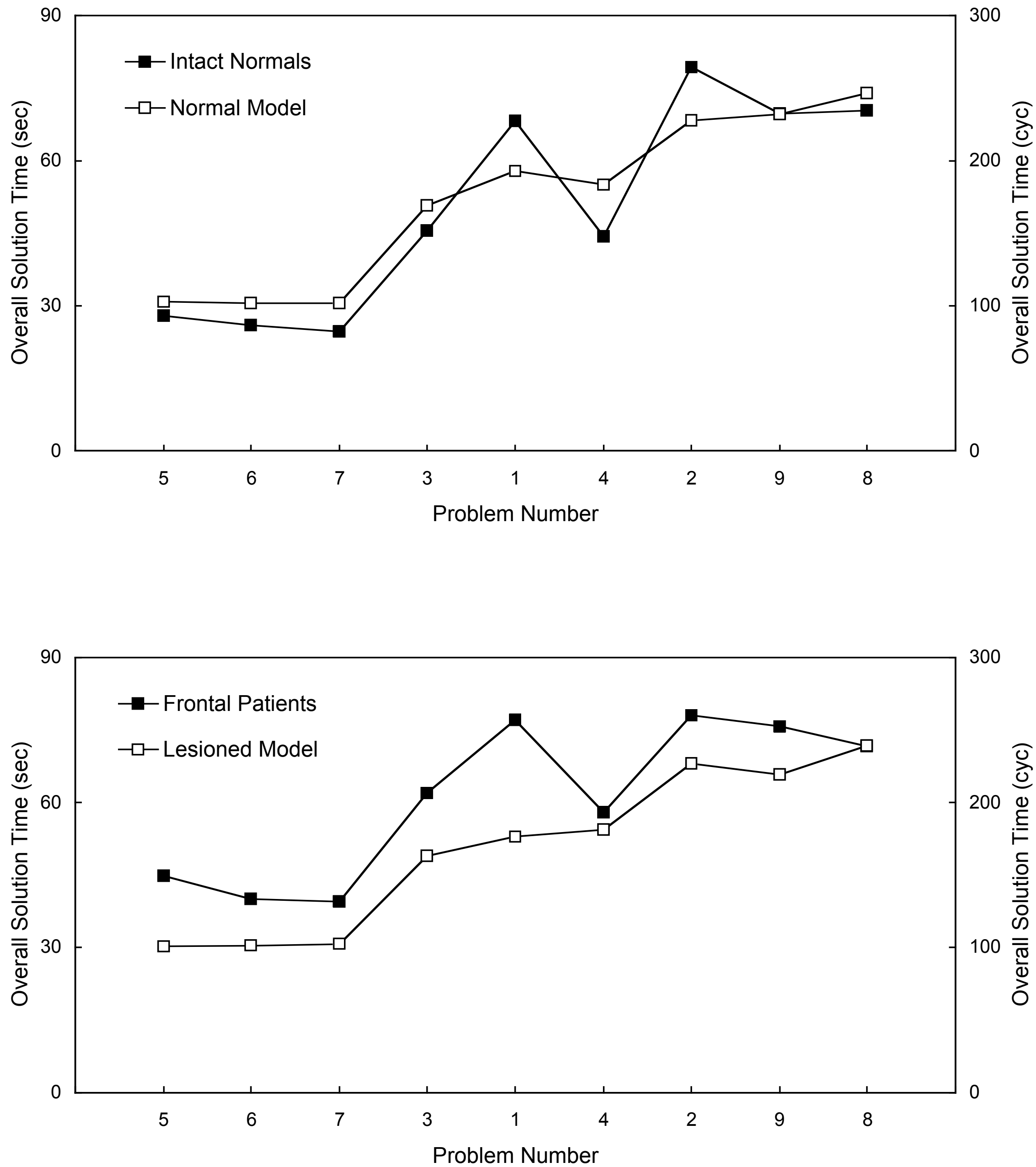


Figure 25. Overall solution time (a) of the intact normals of Goel et al. (2001) and the normal model and (b) of the frontal patients and the lesioned model.

The correlation between the performance of the intact normals and frontal patients and that of the normal and lesioned models is listed in Table 23 for both values of the *top-goal-moves-only* decision. Comparable correlations are achieved regardless of the value of the *top-goal-moves-only* decision ($z=0.08$, $p<0.94$). Table 23 also lists the correlation between the performance of intact normals and frontal patients and that of the normal and lesioned 3CAPS models of Goel et al. (2001). The 4CAPS models where the value of the *top-goal-moves-only* decision is *no* achieve a correlation to the data comparable to that of the 3CAPS models ($z=0.35$, $p<0.73$), as does the 4CAPS model where the value of the *top-goal-moves-only* decision is *yes* ($z=0.49$, $p<0.63$).

Table 23: Correlation between the overall solution time required to solve the Goel et al. (2001) problems by normal and lesioned patients and models for each value of the *top-goal-moves-only* decision.

	Both
4CAPS	
top-goal-moves-only=yes	0.89
top-goal-moves-only=no	0.88
3CAPS	0.84

To summarize, the 4CAPS model has been evaluated against the behavioral data of Goel et al. (2001) on the TOH problem solving of normal young adults and patients with frontal lesions. The resources of LH-Executive were reduced to simulate the effects of left frontal lesions, the resources of RH-Executive were reduced to simulate the effects of right frontal lesions, and the resources of both centers were reduced to simulate the effects of bilateral frontal lesions. The normal and lesioned 4CAPS models provide a good absolute account of the TOH problem solving of intact normals and frontal patients, both in the qualitative sense of similar performance curves and in the quantitative sense of high correlations.

Higher correlations were found when the value of the *top-goal-moves-only* decision was *no* versus *yes* for two of the measures, proportion of problems solved data and number of moves. However, neither advantage was reliable, and this design decision had no impact on the model's fit to the overall solution time data. Therefore, the *top-goal-moves-only* decision remains open.

Finally, consider the relative performance of the 4CAPS models and the 3CAPS models of Goel et al. (2001). The models were statistically indistinguishable across the three measures. Quantitatively speaking, they offer comparable accounts of the data. However, one reason to prefer the 4CAPS model over the 3CAPS model is that it simulates lesions in a more direct and parsimonious way. In the 4CAPS model, lesions are simulated by depleting the resources of the center(s) corresponding to the damaged brain area(s). By contrast, the 3CAPS model requires a chain of auxiliary assumptions to address the lesion data: (1) the frontal lobe is critical for working memory, (2) working memory is critical for TOH problem solving, and therefore (3) frontal lobe lesions can be simulated by reducing the working memory resources available for problem solving. The advantage of the 4CAPS model over the 3CAPS model is not just aesthetic. It also yields predictions about the differentiable effects of left frontal lesions versus right frontal lesions – predictions that can be tested when the appropriate data becomes available. By contrast, the 3CAPS model is silent on questions of lesion laterality.

CHAPTER X

NEUROIMAGING MEASURES OF NORMAL YOUNG ADULTS

The last class of data against which the model will be evaluated consists of neuroimaging data on the TOH problem solving of normal young adults. Until the past decade, information on the neural bases of cognition came primarily from two sources. The first was neuroanatomical and neurophysiological studies of behaving animals. Such data are inherently limited by differences between human and animal brains and by the existence of cognitive tasks that humans can perform but animals cannot, such as language comprehension. The second source of information was behavioral studies of neuropsychological patients. This source suffers from several problems, such as the question of whether the compensatory efforts of patients undermine inferences from their impairments to cognition in intact, normal brains. The recent maturation of neuroimaging technologies such as fMRI has changed all of this. The result has been a flood of relatively direct data on the neural bases of complex cognition in normal young adults.

These data demand formal theoretical accounts, a demand the 4CAPS cognitive architecture was developed to satisfy. Its operating principles were summarized in Chapter V. Briefly, a 4CAPS model consists of multiple centers, each corresponding to a different brain area. Each center possesses a fixed supply of resources to fuel execution of the cognitive functions for which it is specialized. The capacity utilization of a center – the proportion of its resources currently in use – is an instantaneous index of its computational activity. This measure can be averaged over time and compared with the average activation in the corresponding brain area, as measured in neuroimaging studies that employ block designs. It can also be convolved with a hemodynamic response function to produce a predicted activation time series that can be compared with the activation time series observed in the corresponding brain area, as measured in event-related studies.

The primary goal of this chapter is to evaluate the model against the neuroimaging data on the TOH problem solving of normal young adults. A secondary goal is to assess the two remaining design decisions, *top-goal-moves-only* and *suppress-old-states*. This chapter considers the two neuroimaging studies of TOH problem solving by normal young adults that exist in the literature. First, the Fincham et al. (2002) study is reviewed and the reasons why its data could not be modeled are given. Next, the model is evaluated against the Anderson et al. (2005) data. This rich data set includes both behavioral and brain imaging measures collected at the finest of temporal grains.

Fincham, Carter, van Veen, Stenger, and Anderson (2002)

Fincham et al. (2002) had participants solve TOH problems under a constrained presentation paradigm. The problems required between 19 and 23 moves to solve optimally. Embedded in each was one or two instances of a 3-disk tower-to-tower problem. The behavioral data collected were the individual move times and error rates on the 7 moves of the embedded problem. The neuroimaging data collected were the average activations in several regions of interest on low-planning moves, medium-planning moves, and high-planning moves (which require the generation of zero, one, or two new goals, respectively). More activation was found in right DLPFC and bilateral parietal cortex on high-planning moves than on medium-planning and low-planning moves, indicating that these areas are specialized for strategic, goal-driven problem solving. By contrast, left DLPFC exhibited no effect of planning load. These data can potentially be used to evaluate the model's claims about the neural localization of cognitive function.

Unfortunately, there are two reasons why this potential is currently unfulfilled. The first is that the problems Fincham et al. (2002) participants solved do not appear in the published report and were not made available by the corresponding author during the time when this dissertation project was conducted. The second reason is that the behavioral and brain imaging data of interest, which are plotted in figurally in the published report, were not provided by the corresponding author during the time when this dissertation project was conducted.[13] For these reasons, the model could not be applied to the Fincham et al. (2002) data.

Anderson, Albert, and Fincham (2005)

The unavailability of the Fincham et al. (2002) data was tempered by the appearance of a new neuroimaging paper on the TOH problem solving of young adults. Anderson et al. (2005) collected the finest temporal measure, time per individual move, for every move during the solution of complex problems. By contrast, Fincham et al. (2002) focused on just the 7 moves of a simple, embedded problem. Anderson et al. (2005) employed an event-related design, and were therefore able to collect neuroimaging data at the finest temporal resolution currently afforded by fMRI, acquiring one image every 1.5 sec. By contrast, Fincham et al. (2002) acquired one image every 4 sec and averaged activations over moves requiring generation of the same number of new goals (0, 1, or 2). Finally, as a practical point, the problems solved by the Anderson et al. (2005) participants and the numerical values of the plotted data were provided by the corresponding author during the time when this dissertation project was conducted.

The 4CAPS model was evaluated against the data of the first experiment reported in Anderson et al. (2005). (The second and third experiments were not also addressed because they are minor variants of the first.) In this experiment, participants first practiced the sophisticated perceptual strategy over multiple pre-scanner sessions. They then solved isomorphs of the same 5-disk, 28-move problem in the scanner. A constrained presentation paradigm was used throughout. Individual move times were collected, as was an activation time series in three broadly-defined regions of interest. The fit of the model to these data is evaluated below, both in an absolute sense and relative to the fit achieved by the Anderson et al. (2005) ACT-R model. In addition, the neuroimaging data are used to assess the two remaining design decisions.

Behavioral Data

The individual move times of the Anderson et al. (2005) participants are plotted in Figure 26. The model was evaluated against these data. The heuristic weights for simulating the optimal performance observed under constrained presentation paradigms were used. (Recall that these were estimated in Chapter VIII and are listed in Table 11.) The *preference-scheme*, *accrual-scheme*, and *top-moves* decisions were assigned their determined values of *absolute*, *multiplicative*, and *yes*, respectively. As we saw in Chapter VIII, the two remaining design decisions, *top-goal-moves-only* and *suppress-old-states*, have no impact on the model's individual move times, and were therefore assigned arbitrary values of *no* and *all*, respectively. (These design decisions will be evaluated against the neuroimaging data in the next section.) The individual move times of the model are also plotted in Figure 26. Informally speaking, the model does a mediocre job of accounting for the variability of the data. In particular, it displays much greater variability on moves 11 through 15 and 19 through 23 than the human problem solvers.

The result is a 0.68 correlation between the model and data ($z=2.93$, $p<0.004$). Note that this fit is *not* an artifact of the model predicting only two points – the relatively long time for the first move and the relatively short average time for all other moves: when the first move time is omitted, the correlation between the model and data (0.72) is statistically unchanged ($z=0.28$, $p<0.78$).

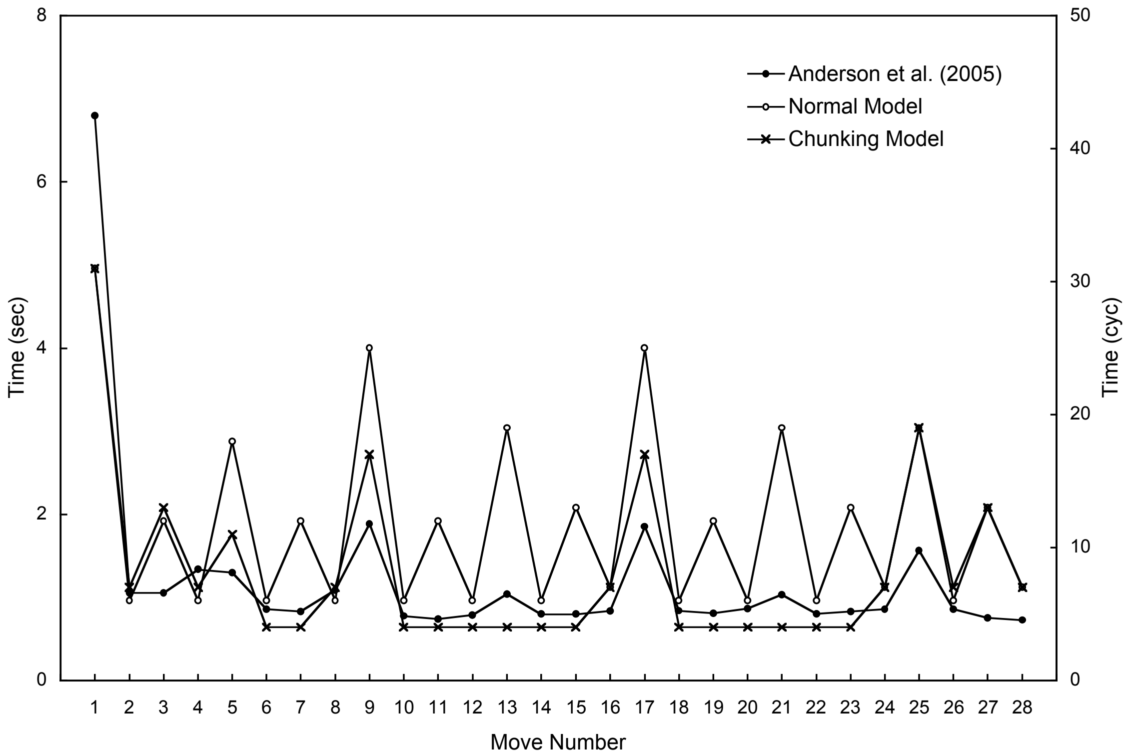


Figure 26. Individual move times of the Anderson et al. (2005) participants and of the normal and chunking models.

Why doesn't the model provide a better account for the Anderson et al. (2005) individual move times? There are two explanations. The first is that the 0.68 correlation is actually not that bad. Recall from Chapter VIII that the correlation between the individual move times of Ruiz (1987) and those of Anderson et al. (1993) is only 0.71, even though the studies sampled from comparable populations, employed the same (constrained) presentation paradigm, and had participants solve 5-disk problems requiring the same number of moves and identical goal operations. This indicates the variability inherent in the individual move times measure. Moreover, this variability was likely inflated by the procedural modifications enacted by Anderson et al. (2005) to compensate for the hostile environment of the scanner: the "Grid of Pittsburgh" isomorph of the TOH task was used to minimize eye movements, participants memorized a puzzle configuration that served as the ending configuration of all problems solved in the scanner, etc. Relative to the variability in the individual move time measure, the 0.68 correlation between the model's individual move times and those of the Anderson et al. (2005) participants is not as bad as it looks at first glance.

There is a second explanation for the mediocre fit between the model and the Anderson et al. (2005) individual move times. The model misses badly on moves 11-15 and 19-23, displaying highly variable move times in contrast to the short, relatively constant move times of the human problem solvers. These 5-move sequences are contained within larger 7-move sequences where naturally-occurring 3-disk tower-to-tower problems are solved. Anderson et al. (2005) suggest that over the multiple pre-scanner practice sessions, participants learned to solve (2-disk and) 3-disk tower-to-tower problems without the use of goals; hence the lack of variation in their move times during these sequences. To evaluate this explanation, a *chunking* version of the 4CAPS model was constructed. Productions were added that propose, without the use of goals, move sequences that solve (2-disk and) 3-disk tower-to-tower problems. The individual move times of the chunking model are also plotted in Figure 26. Informally speaking, it seems to provide a better account of the human data. In particular, the individual move times corresponding to the solution of the embedded 3-disk tower-to-tower problems are relatively short and constant. The chunking model achieves a 0.83 correlation with the data. This is larger than the 0.68 correlation achieved by the normal model, although the difference is not reliable ($z=1.27$, $p<0.21$). Once again, this fit is *not* an artifact of predicting just two points, the relatively long time for the first move and the relatively short average time for all other moves: when the first move time is omitted, the correlation between the chunking model and the data (0.80) is statistically unchanged ($z=0.32$, $p<0.75$).

The 0.83 correlation achieved by the chunking model is lower than the 0.92 correlation achieved by the Anderson et al. (2005) ACT-R model, which also chunks the solution of embedded 2-disk and 3-disk tower-to-tower problems, although this difference is not reliable ($z=1.42$, $p<0.16$). A natural question is whether the chunking 4CAPS model can be modified to match or exceed the fit achieved by the ACT-R model? In my opinion, this is the wrong question to ask. The chunking 4CAPS model is *ad hoc*. There are multiple ways in which the original 4CAPS model can be extended to chunk the solution of 2-disk and 3-disk tower-to-tower problems. The one I chose might not be the one that best fits the individual move time data. However, implementing multiple extensions and anointing as *the* chunking 4CAPS model the one that best fits the data would likely capitalize on chance, i.e., would be unlikely to generalize to other data sets. To avoiding overfitting the individual move time data of Anderson et al. (2005), independent constraints are required on the implementation of chunking. Chief among these is a theory of learning that specifies how the practice regimen of the pre-scanner sessions transforms the original 4CAPS model into a chunking 4CAPS model. Because such constraints are not currently available – the 4CAPS architecture currently lacks learning mechanisms – the chunking model will be set aside and only the original, non-chunking model will be considered for the remainder of this chapter. (The topic of learning is taken up again in Chapter XII.)

Neuroimaging Data

Anderson et al. (2005) employed an event-related design, acquiring an image every 1.5 sec during the solution of each problem. The activation in a particular region of interest for a particular move is that activation in that region in the image that was acquired when the move was made. The activation time series for a particular region of interest, then, is the sequence of activations in that region for the 28 moves required to solve the problem. An activation time series was constructed for a broadly-defined left frontal region, a broadly-defined left parietal region, and a broadly-defined left motor region. The left motor region data will be ignored here because the model contains no center or centers corresponding to any motor area or areas. The activation time series for the left frontal region and that for the left parietal region are plotted in Figure 27. At a macroscopic level, the average activation in the left frontal region is less than the average activation in the left parietal region. At a more microscopic level, there is considerable variation in each activation time series across the 28 moves and 60 sec required to solve the problem. These data constitute a strong test of the temporal dynamics and proposed neural localizations of the 4CAPS model.

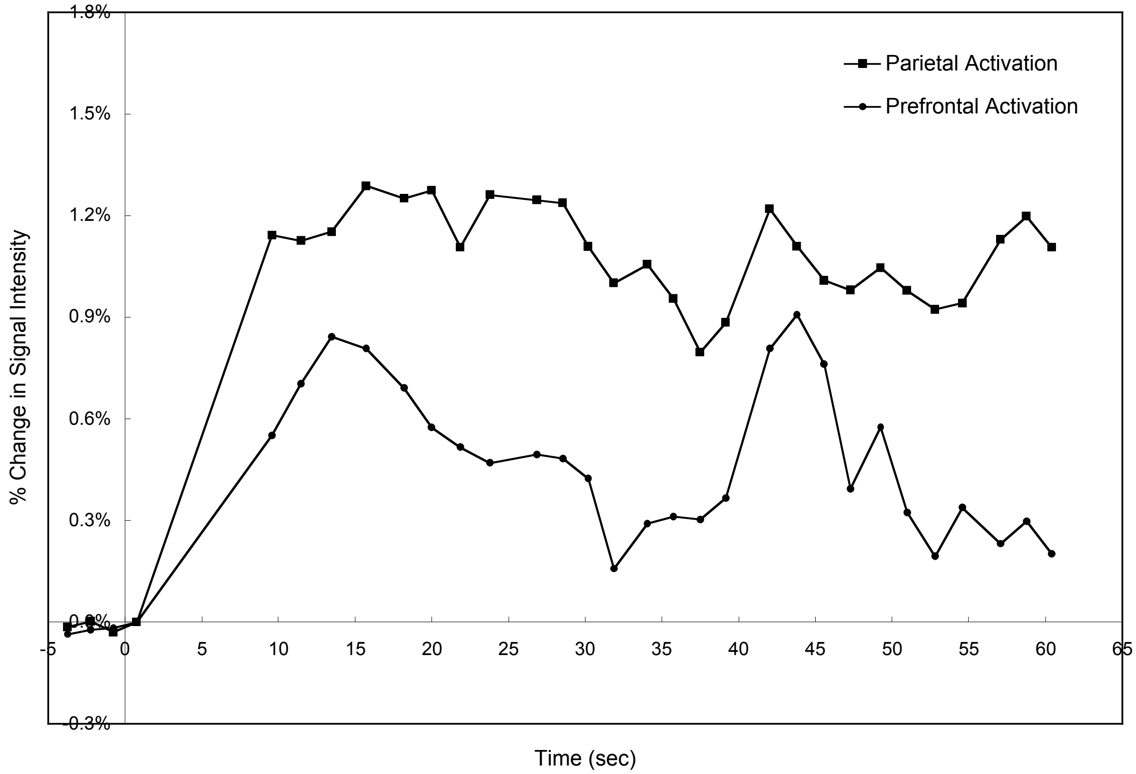


Figure 27. Activation time series for the prefrontal and parietal regions for Anderson et al. (2005).

Two steps were required to fit the 4CAPS model to the Anderson et al. (2005) neuroimaging data. First, four parameters – the resource capacities of the four centers – were estimated. These parameters serve as scaling factors, enabling the model to account for differences in the average activations of different regions of interest. Recall that the smaller the resource capacity of a center, the larger the proportion of its resources consumed during task performance, and thus the larger the predicted activation in the corresponding brain area. Resource capacities for the four centers were identified that produce lower average capacity utilizations in the Executive centers than the Spatial centers, matching the lower average activation observed in the left frontal region versus the left parietal region. These parameter values were fixed for all modeling reported below, and will be discussed no further.

The second step was to account for the hemodynamic response function. Neural computation can be idealized as a sequence of fast impulses. However, the vascular system's response to a neural impulse – the hemodynamic response function – is delayed and distributed in time. The capacity utilization of a model center is an instantaneous measure of the neural computation performed by the corresponding brain area. To compare this measure with event-related fMRI data, the sluggishness of the hemodynamic response must be taken into account. Prior research suggests that this function is well-approximated by a time-delayed gamma function (Aguirre et al., 1998; Boynton et al., 1996) having three parameters: one for the time delay (□) and two for the gamma function (□, *n*). Independent estimates of their values (□=2.5, □=1.25, *n*=3) were taken from the literature.

The Anderson et al. (2005) neuroimaging data were first used to evaluate the two remaining design decisions, *top-goal-moves-only* and *suppress-old-states*. The *top-goal-moves-only* decision specifies the organization of goals in RH-Executive. When its value is *yes*, goals obey a stack-like discipline: only indirect moves that achieve the topmost (i.e., most recent) goal are proposed. When its value is *no*, goals form an unorganized collection: indirect operators are proposed for all goals, and it is left to LH-Executive to sift through them for the one that achieves the topmost goal. The *suppress-old-states* decision governs the suppression of old (i.e., non-current) states in LH-Spatial. When its value is *all*, all old states are suppressed. This represents a maximally efficient policy of freeing resources allocated to all representations not required for future processing. When the value of this decision is *none*, no old states are suppressed. This is a minimally efficient policy given that LH-Spatial, like all centers, possesses a finite resource supply. It is justifiable for certain task goals, such as planning a solution to a problem to be executed at a future time. Finally, when the value of the *suppress-old-states* decision is *non-goal*, all old states are suppressed except for those that led to impasses in problem solving (and ultimately the activation of new goals). This policy is of intermediate efficiency in its utilization of LH-Spatial's resources.

The possible combinations of the *top-goal-moves-only* and *suppress-old-states* decisions yield six model variants. For each variant, the activation time series predicted by each center was correlated with the activation time series observed in the region of interest closest to the brain area corresponding to the center. For example, the activation time series predicted by LH-Executive and that predicted by RH-Executive were compared with that observed in the left frontal region because these centers correspond to areas in the frontal lobe. These correlations were then averaged over model variants for each value of the design decisions; these average correlations are listed in Table 24.

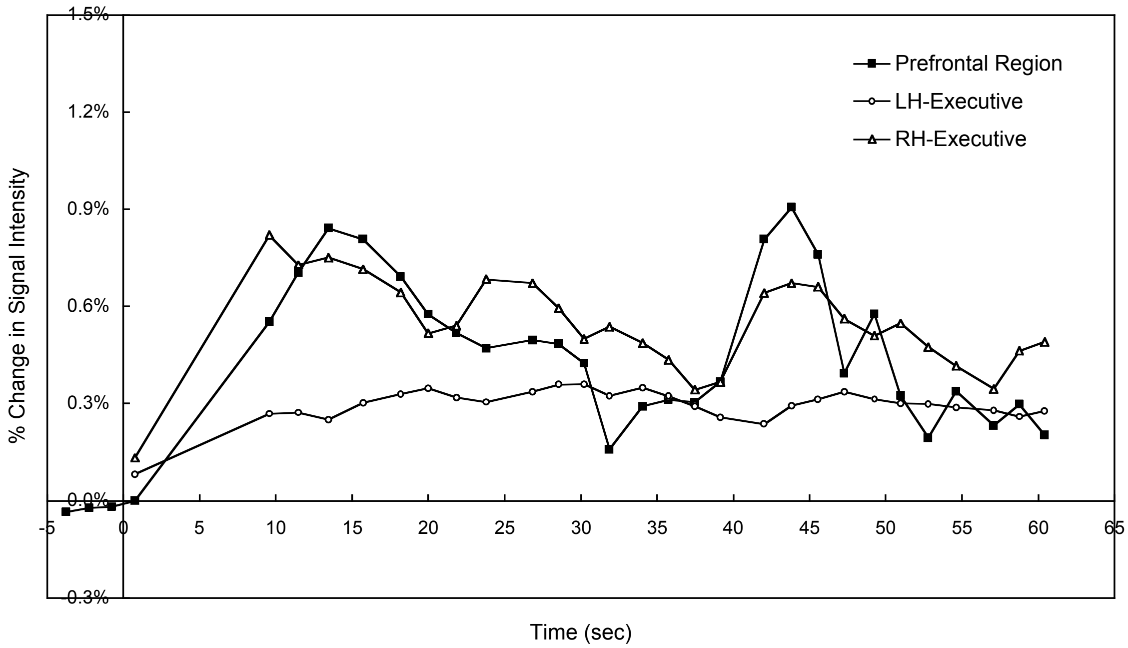
Table 24: Correlation between the activation time series observed in each brain region and predicted by the corresponding model center for the Anderson et al. (2005) problem, averaged for each value of the *top-goal-moves-only* and *suppress-old-states* decisions.

	Value	LH-Executive	RH-Executive	LH-Spatial	RH-Spatial
top-goal-moves-only	*yes*	0.21	0.77	0.72	0.45
	no	0.27	0.78	0.73	0.48
suppress-old-states	*all*	0.24	0.78	0.84	0.46
	non-goal	0.24	0.78	0.77	0.46
	none	0.24	0.78	0.45	0.46

First, consider the *top-goal-moves-only* decision. A value of *no* yields a higher average correlation between LH-Executive and the left frontal region (which contains its corresponding brain area, left DLPFC) than a value of *yes*, although this difference is far from reliable (z=0.23, p<0.82). By contrast, the possible values of the *top-goal-moves-only* decision yield comparable average correlations between RH-Executive and the nearest region of interest, the left frontal region (z=0.14, p<0.89). However, there is a mystery here: The best account of the activations observed in the left frontal region is not the activations predicted by LH-Executive, which corresponds to a brain area contained within this region, but rather the activations predicted by RH-Executive, which corresponds to a brain area contralateral to this region. This is true regardless of whether the value of the *top-goal-moves-only* decision is *no* (z=2.72, p<0.007) or *yes* (z=2.85, p<0.005). We will return to this mystery below. For now, we note that the Anderson et al. (2005) neuroimaging data offer little guidance on the value of the *top-goal-moves-only* decision.

Next, consider the *suppress-old-states* decision. A value of *all* yields a higher average correlation between LH-Spatial and the left parietal region (which contains its corresponding brain area, left IPS) than a value of *non-goal*, although this difference is not reliable (z=0.71, p<0.48), and a higher average correlation than a value of *none*, where this difference is reliable (z=2.60, *p*<0.01). For completeness, the average correlation between LH-Spatial and the left parietal region is higher when the value of the *suppress-old-states* decision is *non-goal* than when it is *none*, although this difference is only marginally reliable (z=1.89, *p*<0.06). Next, consider the RH-Spatial center. The average correlation between RH-Spatial and the nearest region of interest, the left parietal region, is identical regardless of the value of the *suppress-old-states* decision. To summarize, the Anderson et al. (2005) neuroimaging data offer guidance on the *suppress-old-states* decision. They suggest that its value is *all*, they allow that its value might be *non-goal*, and they rule out that its value is *none*. In other words, left IPS, the brain area that corresponds to LH-Spatial, is relatively efficient in suppressing representations that are not required for future processing, freeing its resources for other, more pressing information processing.

We turn now from the assessment of design decisions to the absolute fit of the 4CAPS model to the Anderson et al. (2005) neuroimaging data. The model variant that will be considered embodies the values of the *preference-scheme*, *accrual-scheme*, and *top-moves* decisions determined in previous chapters and the value of the *suppress-old-states* decision determined in the previous paragraph. The *top-goal-moves-only* decision was arbitrarily assigned a value of *no*. The activation time series predicted by each Executive center is plotted in the top panel of Figure 28 alongside the activation time series observed in the left frontal region. The bottom panel plots the activation time series predicted by each Spatial center and the activation time series observed in the left parietal region. Informally speaking the model does a good job. At a macroscopic level, the average activations predicted by the Executive centers are lower than the average activations predicted by the Spatial centers, matching the data. However, it should be noted that this is simply a consequence of the values estimated for four parameters, the resource capacities of the four centers. With a different choice of resource capacities, the model could have just as easily accommodated the opposite (i.e., incorrect) result. To summarize, the model offers no real constraint on the average activations of the left frontal and left parietal regions. At a more microscopic level, the activation time series predicted by each of RH-Executive, LH-Spatial, and RH-Spatial track the waxing and waning of activation observed in the corresponding (or contralateral) brain regions over the 28 moves and 60+ seconds require to solve the 5-disk problem. By contrast, this correspondence is not merely the result of parameter estimation. Rather, it reflects the model's strong claims about the localization of different cognitive functions to different brain areas.



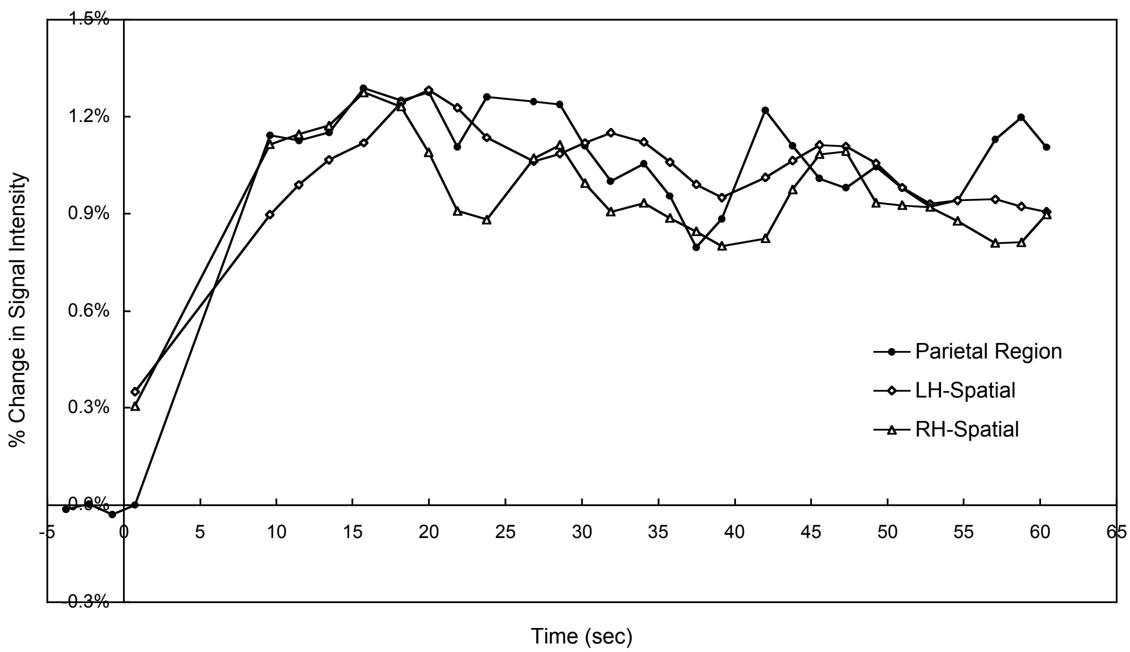


Figure 28. Observed and predicted activation time series (a) for the left frontal region of the Anderson et al. (2005) participants and the Executive centers of the model and (b) for the left parietal region of the Anderson et al. (2005) participants and the Spatial centers of the model.

This can be made more precise. The correlation between the activation time series predicted by each center and the activation time series observed in the region of interest closest to (i.e. containing or contralateral to) the brain area corresponding to that center is listed in Table 25. As we saw above, the activations predicted by RH-Executive correlate more highly with those observed in the left frontal region than the activations predicted by LH-Executive ($z=2.72$, $p<0.007$). In other words, the modulation of left frontal activation appears to be driven more by the goal operations of planning and less by the preferential reasoning of selection. This suggests that the model's localization of these functions to right and left DLPFC, respectively, is exactly backward. This suggestion will be taken up below, in the next paragraph. With respect to the Spatial centers, the activations predicted by LH-Spatial correlate more highly with those observed in the left parietal region than the activations predicted by RH-Spatial ($z=2.47$, $p<0.02$). This suggests that left parietal activation is driven more by the maintenance and transformation of visuospatial representations than it is by the deployment of visuospatial attention. This dissociation corroborates the model's localization of the former function to left IPS.

Table 25: Correlation between the activation time series observed in each brain region and that predicted by the corresponding model center for the Anderson et al. (2005) problem.

Brain Region	LH-Executive	RH-Executive	LH-Spatial	RH-Spatial
Frontal	0.27	0.78	0.73	0.48
Parietal	0.24	0.78	0.84	0.46

Next, consider the mystery described above – the apparent discrepancy between the cognitive functions the model attributes to left and right DLPFC and the activation time series observed in the left frontal region. An obvious inference is that the model gets it exactly backward: that left DLPFC is actually responsible for strategic planning through the articulation of goal hierarchies and right DLPFC for selecting between alternatives via preferential reasoning. If this inference is accepted, and if the functional specializations of LH-Executive and RH-Executive are switched, the result would be an impressively high 0.78 correlation between the activations predicted by LH-Executive and those observed in the left frontal region, and an impressively low 0.27 correlation (i.e., a dissociation) between the activations predicted by RH-Executive and those observed in the right frontal region. However, switching the cognitive functions currently localized to left and right DLPFC would contradict other data on frontal lobe function, such as the Fincham et al. (2002) finding that activation in right DLPFC is an increasing function of planning load (i.e., the number of new goals a move requires), but that no such effect is found in left DLPFC. More generally, it would be consistent with the data that led Milner (1982) to localize short-term memory to right frontal areas and selection between competing responses to left frontal areas.

Another possibility is to take seriously the learning (i.e., chunking) displayed by the Anderson et al. (2005) participants. Recall that their individual move times were relatively short and constant over the intervals of problem solving corresponding to the solution of 3-disk tower-to-tower sub-problems. Anderson et al. (2005) speculated that over multiple pre-scanner sessions, participants chunked the solution of such problems, i.e., learned to solved them without the use of goals. If this is the case, then it likely required the use of long-term declarative memory, which has been localized to temporal areas (e.g., Squire & Zola-Morgan, 1991). These areas are outside the scope of both the normal 4CAPS model and the chunked version described above, which encodes the chunked moves in procedural long-term memory. As Anderson et al. (2005) note, some left frontal areas, including DLPFC and inferior frontal gyrus, have been implicated in the controlled retrieval of information from long-term declarative memory (e.g., Badre & Wagner, 2002). Following this line of reasoning, it might be that LH-Executive correlates poorly with the left frontal region because it does not current implement this retrieval function, which would be critical for predicting activations during moves 9-15 and 17-23, when the embedded 3-disk tower-to-tower problems are solved.

This hypothesis was tested. The correlation between activation time series predicted by LH-Executive and that observed in the left frontal region was computed separately for all 28 moves, for the 14 moves where the Anderson et al. (2005) participants were putatively engaged in normal (non-chunked) problem solving, and for the 14 moves where they were putatively engaged in chunked problem solving. These correlations are listed in Table 26, as are the analogous RH-Executive correlations. They are consistent with the learning story just sketched. Specifically, on the 14 non-chunked moves, where problem solving requires selection between competing moves, a function for which LH-Executive is currently specialized, the correlation to the left frontal region increases to 0.49 from the 0.27 observed on all 28 moves, although this increase is not reliable ($z=0.72$, $p<0.48$). By contrast, on 14 chunked moves, where problem solving requires controlled retrieval from long-term declarative memory, a function for which LH-Executive is not currently specialized, the correlation to the left frontal region is not even positive. An analogous pattern holds for RH-Executive: the correlation to the left frontal region increases to 0.87 on the 14 non-chunked moves from the 0.78 observed on all 28 moves, although this increase is not reliable ($z=0.67$, $p<0.51$), and decreases to 0.62 on the 14 chunked moves, although this increase is also not reliable ($z=0.75$, $p<0.46$). To summarize, the model does not implement the retrieval function, which limits its ability to account for the retrieval of chunked moves from long-term declarative memory, and ultimately suppresses the correlation between LH-Executive and the left frontal region. (The future augmentation of LH-Executive with a controlled retrieval function will be taken up in Chapter XII.)

Table 26: Correlation between the activation time series observed in the left frontal region and predicted by each Executive center for the Anderson et al. (2005) problem, broken down separately for non-chunked and chunked moves.

Data	LH-Executive	RH-Executive
All Moves (28 moves)	0.27	0.78
Non-Chunked Moves Only (14)	0.49	0.87
Chunked Moves Only (14)	-0.42	0.62

Finally, consider the relative fit of the 4CAPS model and the Anderson et al. (2005) ACT-R model to the neuroimaging data. On one hand, the 0.27 correlation between LH-Executive and the left frontal region is smaller that the 0.68 correlation achieved by the ACT-R model ($z=1.95$, $p<0.06$). In addition, the 0.84 correlation between LH-Spatial and the left parietal region is smaller than the 0.92 correlation achieved by the ACT-R model, although this different is also only marginally reliable ($z=1.30$, $p<0.20$). On the other hand, the 0.78 correlation between RH-Executive and the left frontal region is larger than the 0.68 correlation achieved by the ACT-R model, although this center corresponds to a contralateral frontal area and the different is not reliable ($z=0.76$, $p<0.45$). Overall, the 4CAPS model appears to provide a worse fit to the Anderson et al. (2005) neuroimaging data than their ACT-R model.

A closer look qualifies this conclusion. On one hand, the 4CAPS model makes finer-grain predictions than the ACT-R model. In particular, it predicts that activation in right DLPFC should vary dynamically with the number of new goals generated, but that activation in left DLPFC should not (all other things being equal). This prediction is consistent with the Fincham et al. (2002) finding that activation increases in right DLPFC with increasing planning load, but not in left DLPFC. (However, it is inconsistent with the Anderson et al. (2005) data, as we have just seen.) By contrast, the ACT-R model cannot predict differential activation patterns in left and right frontal regions because its predictions concern *bilateral* DLPFC. In this regard, the 4CAPS model potentially offers a finer-grain account of the neural bases of TOH problem solving (e.g., one that can predict different impairments following left versus right frontal lesions) than the ACT-R model – although this potential is currently unrealized. On the other hand, the ACT-R model predicts the activation time series observed in the motor region, a prediction currently outside the scope of the 4CAPS model, which contains no center (or centers) corresponding to motor areas.

CHAPTER XI

GENERAL DISCUSSION

The primary goal of this dissertation project was to develop a computational model of TOH problem solving capable of accounting for and unifying recent empirical work. This goal has been achieved. The model is composed of three distinct levels, each offering different theoretical constraints on the data. The bottom level is the 4CAPS cognitive architecture. It defines the interface between cognitive information processing and cortical information processing. The middle level is a model of fronto-parietal interaction that synthesizes existing theories of problem solving and executive function. Its critical contribution is the definition of problem solving in neurally-localizable terms. The topmost level specializes the general representations and processes of the fronto-parietal model for the TOH domain. It defines the states, operators, goals, and heuristic preferences that implement the sophisticated perceptual strategy. Even with the theoretical constraints offered by the model's three levels, degrees of freedom remain. Some are numerical free parameters in the classic sense; these include the weights of the heuristic productions and the resource capacities of the model centers. Others are design decisions that define the space of model variants.

The model was evaluated against three classes of data. For each class, three questions were asked. First, how well does the model account for the data in the absolute sense of high correlations between model performance and human performance? Second, how good is the model's fit relative to other models of the same data? Third, do the data offer guidance on the design decisions? The model was first evaluated against two behavioral measures collected from normal young adults, time per individual move and number of moves (Anderson et al., 1993; Ruiz, 1987). It provided a good account of the data, both in an absolute sense and relative to competing ACT-R models. Moreover, the data offered guidance on the *preference-scheme*, *accrual-scheme*, and *top-moves* design decisions. The second class of data consisted of three behavioral measures – proportion of problems solved in the allotted time, number of moves, and overall solution time – of patients with frontal lobe lesions (Goel et al., 2001). Patients were simulated by drastically reducing the resources of centers corresponding to damaged brain areas. Once again, the model provided a good account of the data, both in an absolute sense and relative to a competing 3CAPS model. However, these data offered no guidance on either of the two remaining design decisions. The final class of data consisted of neuroimaging measures of normal young adults, specifically activation time series data collected in left frontal and left parietal regions (Anderson et al., 2005). The model again accounted the data well, both in an absolute sense and relative to a competing ACT-R model. Moreover, the neuroimaging data offered guidance on the *suppress-old-states* design decision.

The primary goal of this dissertation project has been largely achieved. The 4CAPS model provides a good account of the new data on TOH problem solving. Although other models offer comparable fits to particular subsets of the data, none approach its empirical breadth. However, accounting for data is just one indicator of a successful computational model. Others include the generation of new theoretical, computational, and methodological insights. This dissertation project has also succeeded in this regard, as described below.

The Nature of Goals

Goals are representations that control cognition over longer periods of time. They are necessary for organizing the performance of complex cognitive tasks. The nature of goals is a long-standing topic in cognitive science. The 4CAPS model provides a novel decomposition of goal operations, spreading their implementation over two different centers, RH-Executive and LH-Executive, corresponding to two different brain areas – right DLPFC and left DLPFC, respectively.

In classical cognitive science, goals are discrete representations organized into stacks. A stack is a computer science data structure that supports two operations: a new representation is stored by the "push" operation, and the topmost (i.e., most recently pushed) representation is retrieved using the "pop" operation. Because the Last representation pushed In is the First item popped Out, stacks are said to obey a LIFO discipline. For several decades, cognitive scientists assumed that goals are stored in perfect stacks, i.e., that are arbitrarily deep and that strictly obey a LIFO discipline. This assumption is part of Soar (Newell, 1990), for example, and is found in all but the most recent versions of ACT (Anderson, 1976; Anderson, 1983; Anderson, 1993).

A perfect goal stack, like a frictionless world, is a simplifying assumption that helps bootstrap a theoretical enterprise but must eventually be replaced by a more realistic analysis. Not surprisingly then, a number of psychologically-plausible implementations of the human goal stack have appeared in recent years. For example, 3CAPS limits the working memory resources available to all representations, including goals, effectively bounding the depth of the goal stack. This results in the loss of goals when the resource limit is reached, which is more likely to happen on relatively difficult tasks and to people with relatively small working memory capacities. Another example is the current version of ACT-R, which stores goals in (and retrieves them from) long-term declarative memory, whose noisy and errorful characteristics result in deviations from perfect LIFO discipline (Altmann & Trafton, 2002; Anderson & Douglass, 2001).

The 4CAPS model of fronto-parietal interaction offers a new decomposition of goal operations. It assumes two kinds of goals. A task goal encodes the task that is current being performed. It establishes the task set, ensuring that all centers “pull in the same direction”. It is the second kind of goal that is of greater interest. These are the conventional goals of cognitive science – what are sometimes called “subgoals”. Goals are stored in RH-Executive, which corresponds to right DLPFC. This center is also credited with the proposal of indirect operators, which strategically achieve goals, and can be contrasted with the direct operators proposed by RH-Spatial, which increase perceptual similarity.

The model’s organization of goals falls short of a perfect stack in three ways. First, only a small number of goals can be stored at any one time. This is because goals are representations in RH-Executive and this center, like all 4CAPS centers, possesses a limited supply of computational resources for maintaining and processing representations. Second, when the value of the *top-goal-moves-only* decision is *no*, goal organization only weakly follows a LIFO discipline. Indirect operators are proposed for all goals in RH-Executive, not just the topmost (i.e., most recent) one. In other words, goals are organized less like a stack and more like an unstructured set.[14] Third, goal organization, the purview of RH-Executive, is separate from goal-based selection, which is the responsibility of LH-Executive. That center’s **prefer-goal** heuristic production directs activation to indirect operators (versus direct operators) and its **prefer-top-goal** heuristic production directs activation to the indirect operator that achieves the topmost goal (versus indirect operators that achieve other goals). Although these heuristic productions bias operator selection in the direction of a LIFO discipline, they do not strictly enforce it, i.e., they do not dominate the biasing of the other heuristic productions. Indirect operators that achieve the topmost goal must still compete against direct operators proposed by RH-Spatial (and, when the value of the *top-goal-moves-only* decision is *no*, against other indirect operators proposed by RH-Executive). This further degrades the LIFO discipline of the goal stack.

This account of the human goal stack differs from previous accounts in a number of ways. Some are relatively simple, such as partitioning operators into separate indirect and and indirect classes. Others are more important, such as decomposing LIFO discipline into separate organization and selection functions and attributing these different functions to different centers (i.e., brain areas). The result is a distributed performance theory of the competence theory of a perfect goal stack, one that admits a distributed cortical implementation. Moreover, this theory can be empirically tested. Consistent with theory, Fincham et al. (2002) found that activation in right DLPFC is modulated by the number of new goals a move requires, but that the same is not the case for left DLPFC. The model also displayed a dissociation between the LH-Executive and RH-Executive centers when simulating the Anderson et al. (2005) data, as Figure 28 shows. (Here, the correspondence between model and human performance was more problematic. The conclusion reached in Chapter X was that a controlled retrieval function is currently missing from LH-Executive, and must be implemented if the model is to account for these data. This is taken up in greater detail in Chapter XII.) In the future, the proposed decomposition of goal operations into separate goal organization and goal-based selection functions and the localization of these functions to different brain areas might be testable by studies of the TOH problem solving of patients with right versus left frontal lesions.

The Nature of Selection

What selects our selections? This question is typically answered by postulating the existence of an executive of the mind, a cognitive CEO – in other words, a homunculus. Invoking a homunculus is traditionally a losing strategy in cognitive science. One contribution of this research project is a new scheme for selection that avoids the well-known defects of homunculi. The seat of selection in the fronto-parietal model is LH-Executive, which corresponds to left DLPC. This center selects among the proposals of other centers (corresponding to other brain areas). For each proposed operator, it first activates a preference to a level commensurate with that operator's absolute heuristic goodness. These activations are then increased multiplicatively until one exceeds threshold. The operator associated with the preference is then selected. (Recall that details of these activation dynamics are specified by the *absolute* value of the *preference-scheme* decision and the *multiplicative* value of the *accrual-scheme* decision.)

This selection scheme combines elements of Newell's (1990) theory of problem solving and Shallice's (1982) theory of executive function as well as some genuinely novel elements. The notion of preferences is taken from Soar. Selection is not via activation of the operators themselves. Operators belong to other centers, and their activations reflect their local goodness in those centers. Selection is via the activation of preferences, which are proxies for operators. Preferences are representations that belong to LH-Executive. The activation of a preference reflects its local goodness in LH-Executive, which is the global goodness of the associated operator relative to all operators proposed by all centers. Preferences are necessary for keeping separate two different notions: the local and global goodness of an operator. Said another way, preferences "tokenize" or "symbolize" the computations of other centers, representing them declaratively, and thus enable LH-Executive to reason over and reconcile them. This is a homunculus, but of a very limited sort: the machinery of selection is exactly the same as that which is being selected, i.e., preferences are just declarative elements and selection heuristics are just productions. This is Newell's fundamental insight about conflict resolution, and it is embodied in the fronto-parietal model.

However, the fronto-parietal model, and in particular the LH-Executive center, is more than a re-implementation of Soar's preference machinery. It has been merged with Shallice's (1982) notion of contention scheduling. The result of this synthesis is two critical changes. First, preferences have been graded with activation dynamics. Preferences accrue activation over time. This is a different style of evidentiary reasoning than Soar's, where preferences are discrete, symbolic representations that are reconciled via a graph-theoretic decision procedure. A second critical change is that preferences are unary: they correspond one-to-one with operators, and their heuristic goodness is a function of just their associated operators. This is a computational improvement of Soar, whose preferences are binary: each asserts that one operator is better than another operator. Binary preferences introduce a number of difficulties. One is that it is possible for inconsistent preferences to be asserted (e.g., *op1* is better than *op2* and *op2* is better than *op1*), producing unnecessary impasses. Another difficulty is that $O(n)$ operators can result in $O(n^2)$ binary preferences. Although this is not a problem for Soar, which has no limits on its computational power, it is for any psychological-plausible cognitive architecture because the number of preferences can quickly exceed the resources available to maintain them.

To summarize, the novel selection scheme developed for the fronto-parietal model softens the preference machinery of Soar to include activation dynamics, or said the other way around, implements Shallice's (1982) notion of contention scheduling using the vocabulary of a leading theory of problem solving. This marriage of the distinct views of Newell and Shallice on selection is an important technical achievement of this dissertation project.

The Reunification of Problem Solving and Executive Function

An important achievement of this dissertation project is the synthesis of two prominent theories in cognitive science, the Soar theory of problem solving (Newell, 1990) and Shallice's (1982) theory of executive function. On one hand, it is surprising that such seemingly dissimilar theories can be unified, especially given that Shallice himself has questioned the scientific contribution of Soar (Cooper & Shallice, 1995). On the other hand, their unification simply reaffirms their shared intellectual ancestry.

Consider first the surprising commensurability of the schemas of Shallice's theory and the (direct) operators of Soar. It can be traced to the common historical antecedents of the two theories, which are not widely known: In his original proposal, Shallice (1982) compared schemas to the productions of Newell and Simon's (1972) classic theory of problem solving. In particular, he compared the perceptual triggers of schemas to the condition sides of productions, both of which are data-driven (pp. 199-200). Given their common ancestry in productions, it is not surprising that the fronto-parietal model manages to align the schemas of Shallice's theory and the (direct) operators of Soar.

The commensurability between the SAS of Shallice's theory and the goal stack of Soar also makes sense from a historical perspective. Shallice's (1982) original description of the SAS relied heavily on analogies to the classic artificial intelligence planning systems of the day, which trace their ancestry back to Newell and Simon's pioneering work on the General Problem Solver (Newell & Simon, 1963; Newell & Simon, 1972; Simon & Newell, 1961). This is easy to forget because the SAS's preferred implementation technology has changed over the years. Norman and Shallice (1986) recast the SAS as a localist connectionist network, perhaps reflecting the influence of the Rumelhart and McClelland (1982) model of word recognition. Shallice (1988) returned to the terminology of symbolic artificial intelligence – this time the MOPs and TOPs of Schank's (1982) theory of dynamic memory. This terminology has stuck, although the implementation technology has since drifted from packets of symbolic information to localist connectionist networks (Cooper & Shallice, 2000). RH-Executive's goal-based implementation of the SAS can be viewed as a return to its computational roots in planning.

It is perhaps most surprising that the preference machinery of Soar and the contention scheduler of Shallice's (1982) theory share intellectual ancestry given their many superficial dissimilarities. Contention scheduling is conceptualized as an activation-based process where schemas compete for selection via lateral inhibition. However, in Shallice's (1982) original proposal, he remarks that "conflict resolution in production systems" is "the model for contention scheduling" (p. 203). Conflict resolution is the process by which one of multiple matching productions is selected for execution. This was hotly debated topic during the 1970s and early 1980s. Many algorithms were proposed, each problematic in its own way. The preference machinery of Soar represents Newell's final solution. In this regard, LH-Executive combines contention scheduling and preferential reasoning in a way that honors the past.

Degrees of Freedom as Design Decisions, not Free Parameters

The TOH model contains two sets of numerically-valued free parameters. The first set consists of the weights of the five heuristic productions in LH-Executive. These weights were estimated twice, once for fitting optimal performance data collected using constrained presentation paradigms and the other for fitting errorful performance data collected using unconstrained presentation paradigms. The second set of free parameters consists of the resource capacities of the four model centers. These capacities were set at large default values when fitting the behavioral data on normal young adults, estimated for the first time when fitting the behavioral data on frontal patients in Chapter IX, and estimated for the second time when fitting the neuroimaging data on normal young adults. With these nine numerical parameters, the model is able to account for different measures collected under different presentation paradigms from different populations. Relative to its broad empirical scope, the model is not over parameterized.

A secondary claim of this dissertation project is that the degrees of freedom of a computational model are not solely in their numerically-valued free parameters, but also in the matrix of design decisions that govern its representations, processes, and control structure. These degrees of computational freedom cannot be evaluated using the same statistical methodology used to evaluate numerically-valued free parameters. A different methodology is required. The one adopted here makes all design decisions visible, varies them orthogonally to define a space of model variants, and prunes this space against the data. The goal was an assessment of each design decision.

The 4CAPS TOH model embodies five design decisions. Two concern LH-Executive, and thus selection among operators proposed by other centers. The *preference-scheme* dictates whether operators are evaluated in an *absolute* sense or *relative* to one another. The *accrual-scheme* specifies whether heuristic activation accumulates to preferences in an *additive* manner, similar to counter models of choice, or in a *multiplicative* manner reminiscent of information-theoretic accounts such as Hick's law. The individual move time data collected from normal young adults offered guidance on these decisions. Specifically, they suggest that LH-Executive's *preference-scheme* is *absolute* and its *accrual-scheme* is *multiplicative*. It should be noted that the data were not definitive in a statistical sense, but rather suggestive in their trends.

The third design decision concerns RH-Spatial. The *top-moves* decision governs the promiscuity with which direct operators are proposed. Recall that direct operators are perceptually-triggered and strive to increase the similarity between the current and ending states. Ending moves are one kind of direct operator that transfer out-of-place disks to their peg positions in the ending configuration. They are always proposed. Top moves are another kind of direct operator. They transfer disks from the tops of pegs to the tops of other pegs (subject to the constraint that larger disks are not placed on top of smaller disks). The *top-moves* decision governs whether top moves are proposed. The individual move time data collected from normal young adults offered guidance on the *top-moves* decision, suggesting that its value is *yes*. Once again, the data were not definitive in a statistical sense, merely suggestive in their trends.

The fourth design decision concerns RH-Executive. The *top-goal-moves-only* decision governs the organization of goals. An open question in cognitive science is whether goals are organized in a stack, and thus follow a LIFO discipline, or whether they form an unstructured set. When the value of this decision is *yes*, goals behave more like a stack. Specifically, the only indirect moves that are proposed are those that achieve the topmost (i.e., most recent) goal. When the value of this decision is *no*, by contrast, indirect moves are proposed for all goals. Unfortunately, the data considered in this dissertation project offered no guidance on the value of the *top-goal-moves-only* decision.

The fifth design decision concerns LH-Spatial. The *suppress-old-states* decision determines the prejudice with which old problem states are suppressed. Recall that LH-Spatial is the visuospatial workspace of the model – the maintainer and transformer of visuospatial representations. Like all centers, it has a fixed supply of resources. Suppressing old states frees resources for other computations. This decision has three possible values: *all* (suppress all old states), *non-goal* (suppress old states that did not produce an impasse, and thus did not spawn goals that are still outstanding), and *none* (suppress no old states). The neuroimaging data collected from normal young adults – specifically, the activation time series observed in the left parietal region that corresponds to LH-Spatial – suggest that the value of the *suppress-old-states* decision is *all*, although they leave open the possibility that it is *non-goal*. Importantly, they rule out the possibility that its value is *none* in a statistically definitive manner.

Validating the 4CAPS Cognitive Architecture

The model is composed of three levels. The bottom level is the 4CAPS cognitive architecture, which defines the interface between cognitive and cortical information processing. The choice of this architecture was critical for the success of this dissertation project, enabling the modeling of both behavioral and neuroimaging data and the simulation of the effects of focal lesions. Conversely, the success of the TOH model has implications for 4CAPS.

The first implication concerns the number of domains 4CAPS covers. As a cognitive architecture, 4CAPS claims to be sufficient for expressing models of all domains of high-level cognition. This is an empirical claim that must be demonstrated on a case-by-case basis. In prior work, it has been demonstrated for the domains of sentence comprehension, TOL problem solving, mental rotation, driving, and complex dual-tasking (Just et al., 1999; Just & Varma, 2006; Newman et al., 2003). TOH problem solving can now be added to this list.

The second implication concerns the empirical measures that 4CAPS models can simulate. The CAPS family of architectures are deterministic, and hence incapable of directly accounting for error measures. Past models have only addressed error data indirectly, e.g., as negative correlations between error rates and the activations of representations reflecting successful task performance. The TOH model takes a first step toward a true account of errorful performance. Specifically, it accounts for two error measures, the number of moves (above the minimum) required to solve problems and the proportion of problems solved in the allotted time. Moreover, it does this for two populations, normal young adults and patients with frontal lesions. This was accomplished by introducing stochasticity at the level of the fronto-parietal model. Specifically, uniformly distributed noise was added to the weights of the five heuristic productions in LH-Executive. This represents the strong claim that the random errors make when performing tasks that engage the fronto-parietal network follow from the stochasticity of a single cognitive function, selecting between proposed operators, that is localized to a single brain area, left DLPFC. This claim has been tested and largely validated against the data on one such task, TOH problem solving. In the future, it must be extended to other tasks that engage the fronto-parietal network, such as TOL problem solving, mental rotation, and driving (Just et al., 1999; Just & Varma, 2006; Newman et al., 2003).

CHAPTER XII

FUTURE DIRECTIONS

This dissertation project has largely achieved its goals. A model has been developed that accounts for a broad range of the new data on TOH problem solving and progress has been made on a number of theoretical, computational, and methodological fronts. This dissertation project also suggests a number of research questions for the future. Three are described below.

Additional Populations

The TOH problem solving of a number of different populations has been documented over the past two decades. This dissertation project has focused on two of these populations, normal young adults and patients with frontal lesions. One goal for future research, then, is to extend the TOH model to additional populations: young children, the elderly, mentally retarded adults, and children diagnosed ADHD. One benefit of doing this will be to broaden the empirical scope of the model's top level, which is specific to the TOH domain. Another benefit will be to better understand the differences between these populations, at least within the context of TOH problem solving.

The TOH task has been used to study the development of cognition in children. Not surprisingly, Piaget (1967) was the first to do so. He found that Stage I children (ages 5.5 to 7) can solve 2-disk problems whereas Stage II children (ages 7.5 to 9) can solve 3-disk problems, an advantage he attributed to their "better subordination of the means to the ends" (p. 295). Klahr and Robinson (1981) conducted the first rigorous experimental study of the development of TOH problem solving. They tested 4, 5, and 6 year old children. Their primary finding was that problem solving improves with age. They also documented a number of interesting second-order effects. More recently, Bishop, Aamodt-Leeper, Creswell, McGurk, and Skuse (2001) investigated the TOH problem solving of 7-8, 9-10, 11-12, and 13-15 year old children. They too found that performance improves with age.

The TOH task has also been used to document the decline of cognition in the elderly. Brennan et al. (1997) had young adult, young elderly, and old elderly participants solve 3-disk and 4-disk problems of varying difficulty. They found a main effect of age as well as an interaction between age and problem difficulty. However, their problem stimuli and the numerical values of their data are no longer available (Brennan, personal communication). The Rönnlund, Lövdén, and Nilsson (2001) study does not suffer from this problem. Their participants ranged from 35 to 85 years in age, and were grouped by five year intervals. They solved the standard 5-disk tower-to-tower TOH problem under a constrained presentation paradigm. Three measures were collected: time per individual move, proportion of participants who solved the problem in the allotted time, and number of illegal moves attempted. They found that across all three measures, performance declines with age.

The problem stimuli used by Klahr and Robinson (1981), Bishop et al. (2001), and Rönnlund et al. (2001) are available as are the numerical values of their data. These studies constitute a solid empirical database against which to evaluate future extensions of the model to the development and decline of TOH problem solving over the human lifespan. Of course, at this point in time, it is unclear which of the model's computational mechanisms will (if in fact any can) bear the explanatory weight. One candidate is the resource capacities of the Executive centers, which might increase with development and decrease with age. Recall that goal operations are attributed to the Executive centers, with goal organization credited to RH-Executive and goal-based selection to LH-Executive. If the resource capacities of these centers increase with development, then so will the number of goals and preferences that can be simultaneously maintained. This will enable the solution of more difficult problems, which effectively require LIFO discipline to be enforced over deep goal stacks. The reverse might be true of the elderly.

A related goal is to extend the TOH model to additional patient populations. Spitz and colleagues have administered the TOH task to mentally retarded adults and to normal children matched on mental age (Borys, Spitz, & Dorans, 1982; Spitz et al., 1982; Spitz, Minsky, & Bessellieu, 1984; 1985). Although the groups are generally comparable on temporal measures of performance, the mentally retarded adults make more moves, especially when solving difficult problems. Another relevant patient population is children diagnosed ADHD. Aman et al. (1998) found that ADHD children require more moves to solve TOH problems than normal, matched controls.

As with the data on cognitive development and aging, it is unclear which of the model's computational mechanisms will be relevant for explaining the results of the Spitz studies (if any) and which will be relevant for the Aman et al. (1998) study (if any). Once again, a reasonable starting point is the resource capacities of the Executive centers. Consistent with this suggestion, Aman et al. (1998) note that ADHD children display reduced blood flow in their frontal lobes and that adults with lifelong untreated ADHD display reduced blood flow in both prefrontal and right posterior parietal areas.

Additional Tasks, Brain Areas, and Functions

At the middle level is a model of the interaction between frontal and parietal areas in the service of complex cognition. TOH problem solving is one such domain, but there are others. Therefore, a natural goal for future research is to extend the fronto-parietal model to additional domains. This goal dovetails with that of continuing to assess the three design decisions that apply to the fronto-parietal model: *preference-scheme*, *accrual-scheme*, and *suppress-old-states*. The data on TOH problem solving offer guidance on the values of these decisions, but except in ruling out the *none* value of the *suppress-old-states* decision, they are not definitive in a statistical sense. Extending the fronto-parietal model to additional tasks will provide new opportunities to assess these decisions.

In fact, this work is already in progress. Earlier versions of the fronto-parietal model were instantiated in the domains of TOL problem solving, mental rotation, and driving. Each of the resulting models were evaluated against the behavioral and brain imaging results of a single study of normal young adults. A logical next step is to reconstruct these models on top of the fronto-parietal model developed for this dissertation project and to evaluate each against a broader range of data. If they provide good accounts of their respective domains, this can be taken as evidence that the fronto-parietal model transcends the TOH task and applies to all tasks that engage frontal and parietal areas.

The fronto-parietal model derives in part from Shallice's (1982) theory of executive function, and it is an interesting question whether it can account for the data on conventional executive function tasks. Executive function tasks can be partitioned into two sets. Tasks belonging to one set require strategic flexibility, and are therefore likely to tap the planning functions of RH-Executive. These include verbal fluency, category fluency, and Wisconsin Card Sorting. Tasks belonging to the other set produce conflict during selection, and are therefore likely to tap the selection function of LH-Executive. These include Stroop, A-not-B, anti-saccade, and flanker. Tasks in the first set are relatively complex and require relatively long periods of time to perform (i.e., tens of seconds), whereas task in the second set are relatively simple and can be performed relatively quickly (i.e., in hundredths of a second). Because the underlying 4CAPS cognitive architecture is geared toward complex cognition, it makes sense to begin with models of executive function tasks belonging to the first set.

For example, consider the verbal fluency task (i.e., "generate words beginning with the letter *f*") and the category fluency task (i.e., "generate the names of animals"). The fluency data suggest a straightforward decomposition of executive function: the number of clusters participants produce indexes controlled strategic processing, whereas the average size of the clusters indexes automatic retrieval from memory (Troyer, Moscovitch, & Winocur, 1997). This functional decomposition explains the performance of lesion patients. In particular, patients with frontal lesions are impaired in their strategic control, and therefore produce fewer clusters, each of normal size, whereas patients with temporal lesions are impaired in their memory retrieval, and therefore produce the normal number of clusters, each of reduced size (Gleissner & Elger, 2001). This functional decomposition is also consistent with the neuroimaging data on the fluency of normal young adults, which suggest that the strategic component is localized to DLPFC and the memorial component to temporal areas, including the hippocampus (Elfgren & Risberg, 1998; Frith, Friston, Liddle, & Frackowiak, 1991; Gourovitch, Kirkby, Goldberg, Weinberger, Gold, Esposito, Van Horn, & Berman, 2000; Phelps, Hyder, Blamire, & Shulman, 1997; Troyer, Moscovitch, Winocur, Alexander, & Stuss, 1998).

Fluency tasks depend heavily on the medial temporal areas where long-term declarative memories are encoded and to the inferior temporal areas where they are stored (e.g., Squire & Zola-Morgan, 1991). Modeling fluency performance will therefore require expanding the fronto-parietal model to also account for fronto-temporal interaction. New centers will have to be constructed that correspond to medial and inferior temporal areas, and the Executive centers will have to be modified to collaborate with them. In particular, LH-Executive will have to be expanded to perform the controlled retrieval function that has been attributed to its corresponding brain area, left DLPFC (e.g., Badre & Wagner, 2002).

An expected benefit of this expansion will be a better account of the Anderson et al. (2005) data. Recall that practice over multiple pre-scanner sessions appears to have enabled their participants to chunk the solution of (2-disk and) 3-disk tower-to-tower problems. One indicator of this was the relatively short and constant move times on the solution of such problems embedded in larger 5-disk problems. This was partially simulated by an *ad hoc* chunking version of the 4CAPS model. Another indicator was the poor fit between the activation time series predicted by LH-Executive, which currently lacks a controlled retrieval function, and that observed in the left frontal region, with the lack of fit most glaring during the solution of the embedded (2-disk and) 3-disk tower-to-tower problems. The development of Memory centers corresponding to medial and inferior temporal areas and the addition of a controlled retrieval function to LH-Executive should enable a better account of the Anderson et al. (2005) data.

A more distal goal of expanding the centers and functions of the fronto-parietal model will be to capture the complex interaction between the strategic/controlled system located in prefrontal areas and the associative/automatic systems located in medial and inferior temporal areas. The strategic/controlled system interprets task instructions, iteratively queries long-term memory, and accumulates the results into a task response. The associative/automatic system responds to retrieval cues with memory traces ordered roughly by their recent frequency in the environment. Said another way, the more distal goal is to explain the interaction between System 2 and System 1 (Kahneman, 2003; Sloman 1996; Stanovich & West, 2000). Such an explanation would lay the ground work for modeling other tasks that depend on the interaction of these systems, such as *ad hoc* categorization (Barsalou, 1983), estimation (Shallice & Evans, 1978), analogy-making (Bunge, Wendelken, Badre, & Wagner, 2005), and tasks that tap the availability heuristic (Tversky & Kahneman, 1973).

Learning

A major lacuna of the TOH model is that it does not learn its solution strategy. The natural approach to filling this gap is to rely on the learning mechanisms of the underlying architecture. This strategy would be to define a base model consisting of two levels: 4CAPS at the bottom and the fronto-parietal model in the middle. The base model would then be seeded with the declarative knowledge of the TOH domain one would expect participants to gain by reading the task instructions. The architecture's learning mechanisms would then be "turned on" and the model run through the same practice regimen that participants experience. The hoped-for result would the automatic acquisition of the topmost, TOH-specific level of the model.

It should be noted that models of TOH problem solving have not generally followed this strategy. For example, Ruiz and Newell (1989/1993) developed a Soar model of TOH problem solving. Its goal recursion strategy was explicitly programmed, not learned by Soar's chunking mechanism. This is also true of the ACT-R models considered above: none learned their solution strategy through that architecture's various symbolic and subsymbolic learning mechanisms.[15]

The one model of TOH problem solving that did learn its solution strategy is, not surprisingly, the connectionist model of Parks and Cardoso (1997). However, this model followed a practice regimen unlike anything experienced by human problem solvers, learned to solve only the standard 3-disk tower-to-tower problem; and barely learned to do even that.

Unfortunately, the TOH model developed for this dissertation project cannot automatically learn its topmost, domain-specific level either. This is because the 4CAPS architecture lacks a learning mechanism. This deficiency, however, can be viewed as an opportunity for future research. Specifically, the TOH model can be used as a testbed for implementing and testing possible 4CAPS learning mechanisms. A modest first step would be to formulate an algorithm for learning the weights of the heuristic productions in LH-Executive. The heuristic productions themselves belong to the fronto-parietal model, and do not have to be learned. However, their weights are free parameters, and therefore perfect candidates for learning. The initial model might assign small random values to these weights. During practice, the weights of productions whose preferences tend to lead to the selection of optimal moves might be increased and the weights of productions whose preferences tend to lead to errorful moves might be decreased. The question is whether a learning algorithm can be formulated that, over a realistic practice regimen, estimates weights reflecting the relative informativeness of each heuristic production. For example, the heuristic production that favors legal operators expresses a generic sort of preference, and its learned weight should therefore be relatively small. By contrast, the heuristic production that favors indirect moves that achieve the topmost goal is critical because it biases selection away from hill-climbing and toward strategic problem solving, and its learned weight should therefore be relatively large.

A more ambitious goal would be to formulate a learning mechanism capable of acquiring new solution strategies, i.e., learning new productions. This would enable an account of the chunking displayed by the Anderson et al. (2005) participants, for example. Recall that they solved TOH problems over multiple practice sessions, and that their individual moves times indicated that they chunked the solution of 2-disk and 3-disk tower-to-tower problems. In other words, they learned productions that propose optimal move sequence for these problems without the need for goals. When the TOH model was supplemented with chunked productions, its fit to these data increased from 0.68 to 0.83. Recall that this chunking model was not considered further because of the *ad hoc* nature of the chunked productions: they were hand-coded, not acquired through a general learning mechanism. This concern would disappear if future research yields a plausible production learning mechanism for 4CAPS.

APPENDIX A: MODEL SOURCE CODE

This appendix contains the source code for the TOH model. Recall that there are actually 48 model variants, one for each combination of the four binary-valued design decisions and the one ternary-valued design decision. The source code can define any one of these variants depending on the values of the *PREFERENCE-SCHEME*, *ACCRUAL-SCHEME*, *TOP-MOVES*, *TOP-GOAL-MOVES-ONLY*, *SUPPRESS-OLD-STATES* variables. To aid the reader, some of the associated code have been grayed out. What remains is the variant that takes the values ABSOLUTE, MULTIPLICATIVE, YES, NO, and ALL for these variables. In addition, recall that a chunked version of the model was described in Chapter X that addresses the effect of practice on TOH problem solving. The source code can also define the chunked model depending on the value of the *CHUNKING* variable. To aid the reader, those aspects of the code specific to the chunked model have been grayed out.

```
(in-package "CL-USER")

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;;         name:  TOH
;;;;      version:  0.5.4
;;;;         date:  8.2005
;;;;
;;;;       author:  Sashank Varma
;;;; organization:  Vanderbilt University
;;;;        email:  sashank.varma@vanderbilt.edu
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; History:
;;;;
;;;;   7.2001 sv: (v0.1.1) First version.  Beginning with the default Fronto-
;;;;                Parietal model centers, specialized the DM classes for the
;;;;                TOH domain, and wrote productions that solve problems that
;;;;                directly solvable, i.e., by perceptual considerations
;;;;                alone).  Made first attempt to layer on the sophisticated
;;;;                perceptual strategy, adding productions that unblock disks
;;;;                that are blocked from above by smaller disks.
;;;;
;;;;   7.2001 sv: (v0.1.2) Perfected the productions for unblocking disks that
;;;;                are blocked from above by smaller disks.
;;;;
;;;;   8.2001 sv: (v0.1.3) Implemented commands for running of batches of
;;;;                simulations and moved them to a separate file TOHGLUE.LSP.
;;;
;;;;   5.2002 sv: (v0.1.4) Fixed three bugs uncovered by Greg Sliwoski and
;;;;                Sharlene Newman.
;;;;
;;;;  11.2004 sv: (v0.2.1) Standardized the comments and the organization of
;;;;                the code.
;;;;
;;;;  11.2004 sv: (v0.2.2) Standardized the Fronto-Parietal model productions
;;;;                across the TOL, mental rotation, driving, and TOH models.
;;;;
;;;;  11.2004 sv: (v0.2.3) Extended the top-level commands for running
;;;;                simulations to display the number of cycles required by
```

```
;;;;                each move so that this measure could be compared with the
;;;;                Ruiz (1987) and Anderson et al. (1993) data.
;;;;
;;;;  11.2004 sv: (v0.2.4) Renamed DM classes and their slots to be more
;;;;                informative and/or concise.  Renamed the TOP-LEVEL-GOAL
;;;;                class to TASK-GOAL; the generic FROM-* and TO-* slot names
;;;;                to the more descriptive SOURCE-* and DEST-*.  Shortened
;;;;                POSITION to POS in several slotnames.  Removed the super-
;;;;                fluous SOURCE-POS slot from goals and moves.
;;;;
;;;;  11.2004 sv: (v0.2.5) Removed binary preferences, leaving only unary
;;;;                preferences, as a prelude to overhauling preference
;;;;                adjudication and operator selection.
;;;;
;;;;  11.2004 sv: (v0.2.6) Implemented four selection schemes that result from
;;;;                two binary-valued design decisions.  The activation accrual
;;;;                scheme is given by the *ACCRUAL-SCHEME* variable, which
;;;;                takes two values, 'MULTIPLICATIVE and 'ADDITIVE.  The
;;;;                preferences scheme is given by the *PREFERENCE-SCHEME*
;;;;                variable, which also takes two values, 'ABSOLUTE and
;;;;                'RELATIVE.  This enables all preference activation
;;;;                productions to be expressed abstractly, at the level of
;;;;                the Fronto-Parietal model.
;;;;
;;;;  12.2004 sv: (v0.2.7) Added the *TOP-MOVES* variable, which takes two
;;;;                values, 'T (yes) and 'NIL (no).  It specifies the
;;;;                promiscuity with which RH-SPATIAL proposes direct moves.
;;;;
;;;;                Made the preference activation productions in LH-EXECUTIVE
;;;;                more selective, replacing OPERATOR condition elements with
;;;;                DIRECT-OPERATOR and INDIRECT-OPERATOR condition elements
;;;;                when possible.
;;;;
;;;;  12.2004 sv: (v0.2.8) Added the *SUPPRESS-OLD-STATES* variable, which
;;;;                takes three values, 'ALL (all old states), 'ALL-BUT-PREV
;;;;                (all but the previous old state), and 'NONE (no old
;;;;                states).  Revised the existing preference activation
;;;;                productions to correctly interact with the possible values
;;;;                of *SUPPRESS-OLD-STATES*.
;;;;
;;;;                An important corollary of this work is that there is now a
;;;;                rough hierarchy of preference activation heuristics: legal,
;;;;                hill-climbing, steepest-hill-climbing, goal, and top-goal.
;;;;
;;;;  12.2004 sv: (v0.2.9) Added the *TOP-GOAL-MOVES-ONLY* variable, which
;;;;                takes two values, 'T (yes) and 'NIL (no).  It specifies
;;;;                whether the goals in RH-EXECUTIVE function more like a
;;;;                stack or more like a set.
;;;;
;;;;                Defined top-level commands to simplify the running of
;;;;                simulations.
;;;;
;;;;  12.2004 sv: (v0.3.1) Renamed the PREFERRED-OPERATOR DM class as SELECTED-
;;;;                OPERATOR in a move away from the terminology of Soar and
;;;;                toward that of the executive function community.
;;;;
;;;;  12.2004 sv: (v0.3.2) Completed the fit to the individual move time data
;;;;                Ruiz (1987) and Anderson et al. (1993) collected from
;;;;                normal young adults.
;;;;
```

```
;;;;  12.2004 sv: (v0.4.2) Completed the fit to the number of moves, proportion
;;;;               of problems solved, and overall solution time Goel et al.
;;;;               (2001) collected from frontal lesion patients and normal
;;;;               controls.
;;;;
;;;;               Eliminated the heuristic production for favoring non-
;;;;               reversal moves because the model behaves the same whether
;;;;               it is included or not.
;;;;
;;;;   1.2005 sv: (v0.4.3) Attempted to fit the Morris et al. (1997) data on
;;;;               frontal lesion patients and normal controls.  However, the
;;;;               stimuli were found to be problematic -- the 5-move
;;;;               congruent problem is actually a conflict problem -- and so
;;;;               the effort was abandoned.
;;;;
;;;;   1.2005 sv: (v0.5.1) Completed the fit to the error rate data Carpenter
;;;;               et al. (1990) collected from normal young adults grouped
;;;;               by fluid intelligence.  This required adding the
;;;;               *CONSTRAINED* variable, which specifies the presentation
;;;;               mode.  When NIL, presentation is unconstrained, and the
;;;;               model is allowed to stray off the optimal path.  When T,
;;;;               presentation is constrained, and the model is constrained
;;;;               to the optimal solution path, which is stored in the
;;;;               *CONSTRAINED-MOVES* variable.  In this case, error rates
;;;;               are tabulated in a kludgy fashion in the variables *ERR0*,
;;;;               *TOT0*, *ERR1*, *TOT1*, *ERR2+*, AND *TOT2+*.
;;;;
;;;;   1.2005 sv: (v0.5.2) Completed the fit to the individual move time data
;;;;               and neuroimaging data Anderson et al. (2005) collected from
;;;;               normal young adults.
;;;;
;;;;               Add a *CHUNKING* variable.  When NIL, the default value,
;;;;               the model runs normally, simulating the performance of
;;;;               a human relatively new to the TOH task.  When T, the model
;;;;               simulates the performance of a human with extensive
;;;;               practice on the TOH task.  In particular, the solution of
;;;;               embedded 2-disk and 3-disk tower-to-tower problems is
;;;;               chunked in productions that do require goal supervision.
;;;;               Chunking increases the model's correlation to the
;;;;               Anderson et al. (2005) individual move time data from
;;;;               0.69 to 0.84 -- within sight of the 0.92 correlation
;;;;               achieved by the ACT-R model.
;;;;
;;;;               Replaced the ALL-BUT-PREV value of the *SUPPRESS-OLD-
;;;;               STATES* variable/design decision, which had been
;;;;               superfluous since the non-reversal selection heuristic was
;;;;               removed (in v0.4.2), with a NON-GOAL value.  Under this
;;;;               design decision, LH-SPATIAL suppresses all old states
;;;;               except those that spawned pending goals.
;;;;
;;;;   3.2005 sv: (v0.5.3) Version of the model described in my original
;;;;               dissertation as submitted 4/2005 and defended 7/2005.
;;;;
;;;;   8.2005 sv: (v0.5.4) Version of the model included as an Appendix in
;;;;               the final version of my dissertation.
;;;;
;;;;               All changes were with an eye toward readability.  Changed
;;;;               all references to *MACRO-CYCS* to *CYCS*, reflecting a
;;;;               change in the 4CAPS interpreter.  Re-organized productions
;;;;               to better match the order in which they are listed in
```

```
;;;;                chapter VII of my dissertation.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; Errata:
;;;;
;;;;   (v0.5.2) Added 0.95 thresholds on the absence tests of all RH-Executive
;;;;     that activate new goals and indirect moves.  I hoped that this would
;;;;     make them iterate in the face of resource shortfalls (like the
;;;;     analogous LH-EXECUTIVE productions), and thus improve the fit to the
;;;;     Carpenter et al. (1990) data and the patient data.  This failed, but
;;;;     I should revisit this in the future.
;;;;
;;;;  (v0.2.3) I was able to get the LH-EXECUTIVE and RH-SPATIAL variants to
;;;;    combine orthogonally.  This was done in part by conditionalizing some
;;;;    of the new LH-EXECUTIVE productions on the value of *PROPOSE-TOP-MOVES*.
;;;;    However, the productions defined when the value of this variable is NIL
;;;;    could just as easily be available all of the time (i.e., regardless of
;;;;    the variables's value).  The only consequence of making this change
;;;;    would in how long the model takes to make the final move when solving a
;;;;    puzzle.  Revisit this in the future.
;;;;
;;;; Consider pervasive binding of DISK and PEG dmes a la the PROPOSE-TOP-MOVE
;;;;   production.
;;;;
;;;; Try to eliminate the DEST-POS slot of the MOVE DM class.
;;;;
;;;; (v0.2.9) Confirmed that among the selection productions in LH-EXECUTIVE,
;;;;   where it makes sense, all DIRECT-OPERATOR CEs could be changed to plain
;;;;   OPERATOR CEs and the model works fine.  This is a more general policy
;;;;   but when the *PREFERENCE-SCHEME* is RELATIVE results in a cumbersomely
;;;;   large number of preferences.
;;;;
;;;; Should ON-TOP-OF-P be true of only the disk immediately on top of the
;;;;   the disk argument or of all disks on top of it?
;;;;
;;;; Eliminate the BASE-GOAL abstraction.
;;;;
;;;; There is an arbitrary tie-breaker in the BUFFER-PEG method.
;;;;
;;;; Determine whether increasing the specialization of RH-EXECUTIVE for the
;;;;   GOAL DM class from 1 to 2 provides a better fit to the neuroimaging
;;;;   data.
;;;;
;;;; Determine whether increasing the specialization of LH-EXECUTIVE for the
;;;;   PREFERENCE DM class from 1 to 2 provides a better fit to the
;;;;   neuroimaging data.
;;;;
;;;; Note that when *PREFERENCE-SCHEME* is 'RELATIVE, then the PREFER-LEGAL
;;;;   heuristic production activates all operators, regardless of their
;;;;   legality.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; Contents:
;;;;
;;;;   (1) Initialization.
;;;;   (2) DM Classes.
;;;;   (3) Centers.
;;;;   (4) The LH-Executive Center.
```

```
;;;;   (5) The RH-Executive Center.
;;;;   (6) The LH-Spatial Center.
;;;;   (7) The RH-Spatial Center.
;;;;   (8) Support Code.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (1) Initialization.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Aspects relevant to the psychological claims of the model.
;;;


;; Design decisions.


; The preference-scheme and accrual-scheme decisions.

;;; There are two different schemes for forming preferences.
;;;      absolute:       Like Shallice's contention scheduler, preferences
;;;                      reflect the absolute goodness of single operators.
;;;      relative:       Like Soar, preferences reflect the relative goodness
;;;                      of pairs of operators.
;;; This design decision is at the level of the Fronto-Parietal model and
;;; affects LH-Executive.
(defvar *preference-scheme* 'absolute)

#|
(setq *preference-scheme* 'absolute)
(setq *preference-scheme* 'relative)
|#

;;; There are two different schemes for accruing activation.
;;;      multiplicative: Like information theory and kind of like Hick's law,
;;;                      increase activation by a constant proportion on every
;;;                      cycle until one operator is above threshold.
;;;      additive:       Like accumulator models of selection, accumulate
;;;                      evidence on every cycle until one operator is above
;;;                      threshold.
;;; This design decision is at the level of the Fronto-Parietal model and
;;; affects LH-Executive.
(defvar *accrual-scheme* 'multiplicative)

#|
(setq *accrual-scheme* 'multiplicative)
(setq *accrual-scheme* 'additive)
|#


; The top-moves decision.
```

```
;;; Governs whether direct moves are proposed promiscuously or conservatively.
;;; This design decision is at the level of the TOH model and affects RH-Spatial.
(defvar *top-moves* t)

#|
(setq *top-moves* t)
(setq *top-moves* nil)
|#


; The top-goal-moves-only decision.

;;; Governs whether indirect moves are proposed promiscuously or conservatively.
;;; This design decision is at the level of the TOH model and affects RH-Executive.
(defvar *top-goal-moves-only* nil)

#|
(setq *top-goal-moves-only* t)
(setq *top-goal-moves-only* nil)
|#


; The suppress-old-states decision.

;;; Governs the prejudice with which old, no longer needed states are suppressed.
;;;      all:      Retain only the current state.
;;;      non-goal: Retain the current state and all previous states that spawned goals.
;;;      nil:      Retain all states.
;;; This design decision is at the level of the Fronto-Parietal model and affects
;;; LH-Spatial.
(defvar *suppress-old-states* 'all)

#|
(setq *suppress-old-states* 'all)
(setq *suppress-old-states* 'non-goal)
(setq *suppress-old-states* nil)
|#


;; Production weights.


; The default weight on productions, i.e., the default rate at which activation
; is propagated.

(defparameter *weight* 1.0)


; Weights for the heuristic productions.

(defvar *w1* 0.0125)     ; legal operators
(defvar *w2* 0.025)      ; hill-climbing operators
(defvar *w3* 0.05)       ; steepest hill-climbing operators
(defvar *w4* 0.05)       ; goal-related operators
(defvar *w5* 0.10)       ; top-goal-related operators

;;; Weight for the iterative activation of preferences when *ACCRUAL-SCHEME* is
;;; MULTIPLICATIVE.
(defvar *w6* 2.0)
```

```
#|
;;; For modeling optimal (i.e., perfect) performance.
(setq *w1* 0.0125)
(setq *w2* 0.025)
(setq *w3* 0.05)
(setq *w4* 0.05)
(setq *w5* 0.10)
(setq *w6* 2.0)

;;; For modeling errorful performance.
(setq *w1* 0.10)
(setq *w2* 0.15)
(setq *w3* 0.05)
(setq *w4* 0.25)
(setq *w5* 0.10)
(setq *w6* 1.0)
|#


; Whether the heuristic weights (e.g., *W1*) are constants (e.g., 0.10) or
; random variables (e.g., U(0, *W1*)).

(defvar *random-weights* nil)

#|
;;; For modeling optimal (i.e., perfect) performance.
(setq *random-weights* nil)

;;; For modeling errorful performance.
(setq *random-weights* t)
|#


;; Whether the model chunks the solution of 2-disk and 3-disk tower-to-tower
;; problems or not.  The chunking model was used to model the results of the
;; Anderson et al. (2005) study, where subjects appear to have learned such
;; a strategy over multiple practice sessions.

(defvar *chunking* nil)

#|
(setq *chunking* nil)
(setq *chunking* t)
|#



;;;
;;; 4CAPS.
;;;


;; Tracing.

(set-tracing-p nil)
(set-tracing-dm-p nil)


;; Activation dynamics.
```

```lisp
;;; The minimum activation level a declarative memory element must possess to
;;; be visible to (i.e., eligible to be matched against) production left-hand
;;; sides.
(set-default-dme-thresh 0.09)



;;;
;;; Fronto-Parietal Model.
;;;


;; Encapsulate whether heuristic production weights are constants or stochastic
;; (i.e., random variables with uniform distributions).

(defun random-weight ()
  (if *random-weights*
    (random 1.0)
    1))



;;;
;;; The TOH Model.
;;;


;; Tracing.

;;; Governs whether the model prints extra tracing information to the terminal
;;; or not.  The extra information documents every move that's made and every
;;; goal that's activated, indented to reflect depth in the goal hierarchy and
;;; time-stamped.
(defvar *toh-tracing* t)

#|
(setq *toh-tracing* t)
(setq *toh-tracing* nil)
|#

(defun goal-indent ()
  (dotimes (n (length (dme-list '(goal))))
    (princ #\tab))
  (values))


;; Whether to record the model's moves during problem solving.

(defparameter *record-moves-p* t)
(defparameter *move-record* ())


;; Constrained problem solving.

;;; Governs whether the model is constrained to the optimal solution path
;;; (stored in *CONSTRAINED-MOVES*) or not.  If it is, then the *ERR* and
;;; *TOT* variables tabulate the frequency with which the model attempted
;;; the wrong move at various goal stack depths.
(defvar *constrained* nil)
```

```
#|
(setq *constrained* nil)
(setq *constrained* t)
|#

(defvar *constrained-moves* ())

(defparameter *err0* 0)
(defparameter *tot0* 0)
(defparameter *err1* 0)
(defparameter *tot1* 0)
(defparameter *err2+* 0)
(defparameter *tot2+* 0)




;;;
;;; Visuospatial representations.
;;;
;;; The Fronto-Parietal model does not completely specify the nature of
;;; visuospatial representations located in parietal areas.  Rather, it assumes
;;; that visuospatial representations can be encapsulated into states.  This
;;; section defines the visuospatial representation of TOH puzzle configurations
;;; and their elements (disks, pegs, positions) functionally.  Productions are
;;; written to this functional interface, which abstracts over the details of
;;; their implementations in Lisp.  For example, a production left-hand side
;;; can test whether a particular disk is on top of its peg using the TOP-DISK-P
;;; predicate without any knowledge of how visuospatial representations are
;;; are implemented (i.e., as lists, arrays, bitmaps, etc.).  This prevents
;;; implementation details from creeping into the centers, productions, and
;;; declarative memory elements of the model, keeping them psychologically
;;; clean.  It also enables the programmer to change the implementation in the
;;; future without compromising the rest of the model.
;;;
;;; It is recommended that the programmer new to this code skip over the
;;; details of how predicates such as TOP-DISK-P are implemented in Lisp.
;;;


;; Map between symbolic and numeric peg designators.

(defun peg-number (peg)
  (ecase peg
    (peg1 1)
    (peg2 2)
    (peg3 3)))

(defun number-peg (num)
  (ecase num
    (1 'peg1)
    (2 'peg2)
    (3 'peg3)))


;; Return the buffer peg given source and destination pegs.

(defun other-peg (p1 p2)
  (number-peg (- 6 (peg-number p1) (peg-number p2))))
```

```
;; The puzzle configuration class.

(defclass configuration ()
  ((disks :initarg :disks
          :initform nil
          :accessor disks)
   (peg1 :initarg :peg1
         :initform nil
         :accessor peg1)
   (peg2 :initarg :peg2
         :initform nil
         :accessor peg2)
   (peg3 :initarg :peg3
         :initform nil
         :accessor peg3)))


; Basic methods.

(defmethod print-object ((self configuration) str)
  (format str "~A ~A ~A" (peg1 self) (peg2 self) (peg3 self)))

(defmethod copy-config ((self configuration))
  (make-instance 'configuration
    :disks (disks self)
    :peg1 (copy-list (peg1 self))
    :peg2 (copy-list (peg2 self))
    :peg3 (copy-list (peg3 self))))


; Moves.

(defmethod make-move ((self configuration) n dp)
  (let ((sp (peg-of self n)))
    (setf (slot-value self sp) (delete n (slot-value self sp)))
    (setf (slot-value self dp) (nconc (slot-value self dp) (list n))))
  self)

(defmethod try-move ((self configuration) n dp)
  (make-move (copy-config self) n dp))


; Pegs.

(defmethod peg-of ((self configuration) n)
  (assert (<= n (disks self)))
  (or (find n '(peg1 peg2 peg3) :test #'(lambda (x y)
                                          (member x (slot-value self y))))
      (error "Disk ~A not on any peg of ~A." n self)))

;;; Returns first non-empty peg.
(defmethod peg-with-disks ((self configuration))
  (cond ((peg1 self)
         'peg1)
        ((peg2 self)
         'peg2)
        ((peg3 self)
         'peg3)))
```

```
(defmethod random-other-peg ((self configuration) n)
  (let ((sp (peg-of self n)))
    (random-element (remove-if #'(lambda (dp)
                                   (or (eq dp sp)
                                       (destination-blocking-disks self n dp)))
                               '(peg1 peg2 peg3)))))


; Positions.

(defmethod position-of ((self configuration) n)
  (position n (slot-value self (peg-of self n))))

(defmethod top-empty-position ((self configuration) peg)
  (length (slot-value self peg)))

(defmethod occupied-position-p ((self configuration) peg pos)
  (< pos (top-empty-position self peg)))


; Disks.

(defmethod random-top-disk ((self configuration))
  (let ((disks ()))
    (let ((disk1 (first (last (peg1 self)))))
      (when disk1
        (setq disks (nconc disks (list disk1)))))
    (let ((disk2 (first (last (peg2 self)))))
      (when disk2
        (setq disks (nconc disks (list disk2)))))
    (let ((disk3 (first (last (peg3 self)))))
      (when disk3
        (setq disks (nconc disks (list disk3)))))
    (random-element disks)))

(defmethod on-top-of-p ((self configuration) n1 n2)
  (and (eq (peg-of self n1) (peg-of self n2))
       (= (position-of self n1)
          (1+ (position-of self n2)))))

(defmethod top-disk-p ((self configuration) n)
  (or (eql n (first (last (peg1 self))))
      (eql n (first (last (peg2 self))))
      (eql n (first (last (peg3 self))))))

(defmethod blocked-disk-p ((self configuration) n)
  (not (top-disk-p self n)))


; Out-of-place disks.

(defmethod out-of-place-disks ((curr configuration) (end configuration))
  (let ((out-of-place-disks ()))
    (do ((disk (disks curr) (1- disk)))
        ((zerop disk))
      (unless (eq (peg-of curr disk) (peg-of end disk))
        (push disk out-of-place-disks)))
    out-of-place-disks))
```

```
(defmethod random-out-of-place-disk ((curr configuration) (end configuration))
  (random-element (out-of-place-disks curr end)))

(defmethod largest-out-of-place-disk ((curr configuration) (end configuration))
  (let ((disks (out-of-place-disks curr end)))
    (and disks (apply #'max disks))))

(defmethod buffer-peg-position ((self configuration) peg n)
  (let ((disks (slot-value self peg)))
    (or (position-if #'(lambda (disk)
                         (< disk n))
                     disks)
      (length disks))))


; Disks blocking source pegs.

(defmethod source-blocking-disks ((self configuration) n)
  (rest (member n (slot-value self (peg-of self n)))))

(defmethod random-source-blocking-disk ((self configuration) n)
  (random-element (source-blocking-disks self n)))

(defmethod largest-source-blocking-disk ((self configuration) n)
  (first (source-blocking-disks self n)))


; Disks blocking destination pegs.

(defmethod destination-blocking-disks ((self configuration) n dp)
  (do ((disks (slot-value self dp) (rest disks)))
      ((or (null disks)
           (> n (first disks)))
       disks)))

(defmethod random-destination-blocking-disk ((self configuration) n dp)
  (random-element (destination-blocking-disks self n dp)))

(defmethod largest-destination-blocking-disk ((self configuration) n dp)
  (first (destination-blocking-disks self n dp)))


; Blocking disks more generally.

(defmethod blocking-disks-p ((self configuration) n dp)
  (or (source-blocking-disks self n)
      (destination-blocking-disks self n dp)))

(defmethod largest-blocking-disk ((self configuration) n dp)
  (let ((source-block (largest-source-blocking-disk self n))
        (dest-block (largest-destination-blocking-disk self n dp)))
    (if source-block
      (if dest-block
        (max source-block dest-block)
        source-block)
      dest-block)))


; Auxiliary function.
```

```
(defun random-element (lis)
  (nth (random (length lis)) lis))




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (2) DM classes.
;;;;
;;;; In production system languages, representations take the form of
;;;; declarative memory elements, which are structures with names components.
;;;; Components are called "slots" or "attributes" or "features".  What slots
;;;; a dme has are governed by its class.  This section defines the DM classes
;;;; of the model.  These classes behave like the classes of object-oriented
;;;; programming languages.  Associated with each class are a number of
;;;; methods, sometimes called "virtual functions."  Subclasses are derived
;;;; from superclasses, inheriting their slots and methods, which they can
;;;; optionally override.
;;;;
;;;; The DM class hierarchy is represented schematically below, where "A: B (+C)"
;;;; means that class B is a subclass of class A, and also optionally class C,
;;;; which is called a "mixin" but is called other things in object-oriented
;;;; programming languages that do not support multiple inheritance.  (For
;;;; example, Java simulates mixins with "interfaces".)
;;;;
;;;; DM classes that belong to the Fronto-Parietal model are capitalized;
;;;; those specific to the TOH model are written in lower-case letters.
;;;;
;;;; puzzle-mi
;;;;
;;;; move-mi
;;;;
;;;; BASE-STATE:          STATE:              puzzle           (+ puzzle-mi)
;;;;                      END-STATE:          end-puzzle       (+ puzzle-mi)
;;;;
;;;; OPERATOR:            DIRECT-OPERATOR:    direct-move      (+ move-mi)
;;;;                      INDIRECT-OPERATOR:  indirect-move    (+ move-mi)
;;;;
;;;; PREFERENCE:
;;;;
;;;; SELECTED-OPERATOR
;;;;
;;;; BASE-GOAL:           TASK-GOAL:          solve-puzzle-goal
;;;;                      GOAL:               unblock-goal
;;;;
;;;; disk
;;;;
;;;; peg
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; Note: For the chunked version of the model, which uses chunked sequences
;;;;       of indirect moves to solve embedded 2-disk and 3-disk tower-to-tower
;;;;       problems without the use of goals, the DM hierarchy also includes the
;;;;       following classes:
;;;;
;;;; [...] indirect-move:  chunked-indirect-move
;;;;
;;;; [...] unblock-goal:   unblock-disk-goal:      chunked-unblock-disk-goal
```

```
;;;;                        unblock-position-goal:  chunked-unblock-position-goal
;;;;
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal Model.
;;;


;; STATE-related classes and methods.


; The BASE-STATE class from which all others inherit.  This abstract class
; is convenient because STATE and END-STATE classes typically share the same
; contents, which can be defined once, here.

(defdmclass base-state ()
  contents)


; The STATE class, instances of which are the nodes in Newell and Simon's
; (1972) view of problem spaces as graphs.

(defdmclass state (base-state))

(defmethod more-recent-state-p ((s1 state) (s2 state))
  (> (id s1) (id s2)))


; The END-STATE class, which represents task completion.

(defdmclass end-state (base-state))


;; STATE-related multimethods.  These methods apply to more than one STATE
;; class.

;;; Are the two states S1 and S2 the same?  When the Fronto-Parietal model is
;;; instantiated in a particular domain, this method must be overridden with one
;;; sensitive to the content of states in that domain: puzzle configurations in
;;; Tower of London, Shepard-Metzler figures in mental rotation, the road view
;;; in driving, and puzzle configurations in Tower of Hanoi.
(defmethod contents-equal ((s1 t) (s2 t))
  (error "The CONTENTS-EQUAL multimethod must be specialized for the task at hand."))

;;; Is the state S, typically the current state, the same as the end state ES?
;;; This determines whether the current task has been completed.
(defmethod not-solved-p ((s state) (es end-state))
  (not (contents-equal (contents s) (contents es))))


;; The OPERATOR class, which in Newell and Simon's (1972) theory of problem
;; solving are the means for moving between the states of the problem space.
;; When applied to a current state, the result is the next state.

(defdmclass operator ()
  state)
```

```
(defdmclass direct-operator (operator))

(defdmclass indirect-operator (operator)
  goal)


;; The PREFERENCE class.  In Newell's (1990) Soar theory of problem solving,
;; multiple operators can apply to the current state.  This ambiguity is
;; resolved through the assertion of preferences, which represent the relative
;; goodness of operations.  They are the means by which the next operator to
;; apply is selected.

(defdmclass preference ()
  operator)


;; The SELECTED-OPERATOR class.  In Newell's (1990) Soar theory of problem
;; solving, when one preference accumulates enough heuristic activation to be
;; above threshold, an instance of this class marks it as selected.

(defdmclass selected-operator ()
  operator)


;; GOAL-related classes and methods.


; The BASE-GOAL class from which all others inherit.  This can be a convenient
; abstraction when the GOAL and TASK-GOAL classes share structure, which can be
; defined once, here.

(defdmclass base-goal ())

(defmethod more-recent-goal-p ((bg1 base-goal) (bg2 base-goal))
  (> (id bg1) (id bg2)))


; The TASK-GOAL class, which is subclassed to represent the domain-specific
; task being performed.  This operationalizes the notion of "task set," and can
; come in useful when modeling dual-tasking.

(defdmclass task-goal (base-goal))


; The GOAL class, which is specialized to represent the subgoals that are
; useful for resolving the impasses that arise in particular domains.  Goals
; can be proposed recursively, producing the goal-subgoal hierarchies central
; to Newell and Simon's (1972) theory of problem solving.

(defdmclass goal (base-goal)
  operator)


;; Multimethods that apply across classes of the Fronto-Parietal model.  They
;; must be overridden with ones sensitive to the vagaries of the domain to
;; which the model is being applied.

;;; Does operator OP1 produce a greater resemblance to the end-state ES than
;;; OP2?
```

```lisp
(defmethod steeper-climbing-operator-p ((op1 operator) (op2 operator) (es end-state))
  (error "The STEEPER-CLIMBING-OPERATOR-P multimethod must be specialized for the task."))

;;; Apply operator OP to state S, yielding the contents of the resulting state.
(defmethod perform-operator ((op operator) (s state))
  (error "The PERFORM-OPERATOR multimethod must be specialized for the task."))

;;; Does the state S indicate the attainment of goal G?  (In other words, has
;;; the impasse that prompted goal G been resolved?)
(defmethod satisfied-p ((g goal) (s state))
  (error "The SATISFIED-P multimethod must be specialized for the task."))

;;; Can operator OP be applied to state S, i.e., are all of its preconditions
;;; satisfied?
(defmethod legal-operator-p ((s state) (op operator))
  (error "The LEGAL-OPERATOR-P multimethod must be specialized for the task."))



;;;
;;; TOH Model.
;;;


;; Support classes.

(defdmclass disk ()
  disk)

(defdmclass peg ()
  peg)


;; Mixin classes.  Often, two classes that reside in different branches of the
;; inheritance hierarchy will share slots and methods.  In these cases, it is
;; useful to extract the shared components in separate "mixin" classes.


;

(defdmclass puzzle-mi ())


;

(defdmclass move-mi ()
  disk
  source-peg
  dest-peg
  dest-pos)



;; STATE-related classes.

(defdmclass puzzle (puzzle-mi state))

(defdmclass end-puzzle (puzzle-mi end-state))
```

```
;; OPERATOR-related classes.

(defdmclass direct-move (move-mi direct-operator))

(defdmclass indirect-move (move-mi indirect-operator))


;; GOAL-related classes and methods.

(defdmclass solve-puzzle-goal (task-goal))

(defdmclass unblock-goal (goal)
  source-peg
  dest-peg
  dest-pos)


; These subgoals are specified by the sophisticated perceptual strategy that
; the model adopts.

(defdmclass unblock-disk-goal (unblock-goal)
  disk)

(defdmclass unblock-position-goal (unblock-goal))


;

(defmethod buffer-peg ((g goal))
  (with-slots (source-peg dest-peg) g
    (if (eq source-peg dest-peg)
      (if (eq source-peg 'peg1)     ; Arbitrary tie-breaker.
        'peg2
        'peg1)
      (other-peg source-peg dest-peg))))


;; Classes defined only the for the chunking version of the model, which solves
;; embedded 2-disk and 3-disk tower-to-tower problems without the use of goals.

(defdmclass chunked-indirect-move (indirect-move))

(defdmclass chunked-unblock-disk-goal (unblock-disk-goal))

(defdmclass chunked-unblock-position-goal (unblock-position-goal))



;;;
;;; TOH Model multimethods.
;;;
;;; Override the general multimethods of the Fronto-Parietal model with the
;;; details of the TOH domain.
;;;


;;

(defmethod contents-equal ((c1 configuration) (c2 configuration))
  (and (equal (peg1 c1) (peg1 c2))
```

```
      (equal (peg2 c1) (peg2 c2))
      (equal (peg3 c1) (peg3 c2))))


;;

(defmethod hill-climbing-operator-p ((m move-mi) (ep end-puzzle))
  (and (eql (peg-of (contents ep) (disk m))
            (dest-peg m))
       (eql (position-of (contents ep) (disk m))
            (dest-pos m))))

(defmethod steeper-climbing-operator-p ((m1 move-mi) (m2 move-mi) (ep end-puzzle))
  (and (hill-climbing-operator-p m1 ep)
       (hill-climbing-operator-p m2 ep)
       (> (disk m1) (disk m2))))


;;

(defmethod perform-operator ((m move-mi) (p puzzle))
  (if *constrained*
    (if *constrained-moves*
      (let ((move (first *constrained-moves*)))
        (make-move (copy-config (contents p))
                   (second move)
                   (third move)))
      (error "Model continued to run, possibly after problem solved."))
    (make-move (copy-config (contents p))
               (disk m)
               (dest-peg m))))

;;

(defmethod satisfied-p ((g unblock-disk-goal) (p puzzle))
  (top-disk-p (contents p) (disk g)))

(defmethod satisfied-p ((g unblock-position-goal) (p puzzle))
  (eql (dest-pos g)
       (top-empty-position (contents p) (dest-peg g))))

(defmethod satisfied-p ((g chunked-unblock-disk-goal) (p puzzle))
  (and (= (disk g) 3)
       (eq (peg-of (contents p) 3) (dest-peg g))
       (on-top-of-p (contents p) 2 3)
       (on-top-of-p (contents p) 1 2)
       (top-disk-p (contents p) 1)))

(defmethod satisfied-p ((g chunked-unblock-position-goal) (p puzzle))
  (and (eq (peg-of (contents p) 2) (buffer-peg g))
       (on-top-of-p (contents p) 1 2)
       (top-disk-p (contents p) 1)))


;;

(defmethod legal-operator-p ((c configuration) (m move-mi))
  (and (top-disk-p c (disk m))
       (= (dest-pos m) (top-empty-position c (dest-peg m)))
       (or (null (slot-value c (dest-peg m)))
```

```
           (< (disk m) (first (last (slot-value c (dest-peg m))))))))



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (3) Centers.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal Model.
;;;


;; Delete all existing centers.

(del-centers)


;; Add the centers of the Fronto-Parietal model.

(add-center support)

(add-center rh-executive)
(add-center lh-executive)

(add-center rh-spatial)
(add-center lh-spatial)


;; As the default, give the centers unlimited resource supplies.

(set-caps@ (support rh-executive lh-executive rh-spatial lh-spatial) nil)


;; Define the functional specializations of the centers.
;;
;; In 4CAPS, 1 represents perfect specialization, a number greater than 1
;; less-that-perfect specialization, T read-only access to dmes of that
;; (sub)class, and NIL not even access.

(set-specs@ lh-executive base-dme nil
                         base-state t
                         operator t
                         preference 1
                         selected-operator 1
                         base-goal t)

(set-specs@ rh-executive base-dme nil
                         base-state t
                         operator t
                         preference t
                         selected-operator t
                         base-goal 1)

(set-specs@ lh-spatial base-dme nil
```

```
                       base-state 1
                       operator t
                       preference t
                       selected-operator t
                       base-goal t)

(set-specs@ rh-spatial base-dme nil
                       base-state t
                       operator 1
                       preference t
                       selected-operator t
                       base-goal t)



;;;
;;; TOH model.
;;;


;; Set the resource capacities of the centers.

(set-caps@ support nil)

(set-caps@ lh-executive 10.0)
(set-caps@ rh-executive 10.0)

(set-caps@ lh-spatial 10.0)
(set-caps@ rh-spatial 10.0)


;; Define the specializations of the centers for the functions specific to the
;; TOH domain.

(set-specs@ support base-dme nil
                    disk 1
                    peg 1)

(set-specs@ lh-executive disk t
                         peg t)

(set-specs@ rh-executive disk t
                         peg t
                         indirect-move 1
                         solve-puzzle-goal 1
                         unblock-goal 1)

(set-specs@ lh-spatial disk t
                       peg t)

(set-specs@ rh-spatial disk t
                       peg t
                       direct-move 1
                       indirect-move t)



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
```

```
;;;; (4) The LH-Executive Center.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal model productions.
;;;


;; Selection heuristic productions


; Prefer legal operators.

(cond ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'absolute))
       (when *top-moves*

         (p@ lh-executive prefer-legal-aa ((s state)
                                          (op direct-operator))
             (equal s (state op))
             (legal-operator-p (contents s) op)
             (*no ((~s state))
               (more-recent-state-p ~s s))
             (*no ((~pr preference 0.95))
               (equal (state (operator ~pr)) s))
             (*no ((~sop selected-operator))
               (equals (state (operator ~sop)) s))
         -->
             (spew t (preference :operator op)
                     (* *w1* (random-weight)))
         )

         ))
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'absolute))
       (when *top-moves*

         (p@ lh-executive prefer-legal-ma ((s state)
                                          (op direct-operator))
             (equal s (state op))
             (legal-operator-p (contents s) op)
             (*no ((~s state))
               (more-recent-state-p ~s s))
             (*no ((~pr preference 0.01))
               (equal (state (operator ~pr)) s))
             (*no ((~sop selected-operator))
               (equals (state (operator ~sop)) s))
         -->
             (spew t (preference :operator op)
                     (* *w1* (random-weight)))
         )

         ))
      ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'relative))
       (if *top-moves*
```

```
  (p@ lh-executive prefer-legal-ar1 ((s state)
                                     (op1 direct-operator)
                                     (op2 direct-operator))
      (equals s (state op1) (state op2))
      (legal-operator-p (contents s) op1)
      (*whole (not (legal-operator-p (contents s) op2)))
      (*no ((~s state))
        (more-recent-state-p ~s s))
      (*no ((~pr preference 0.95))
        (equal (state (operator ~pr)) s))
      (*no ((~sop selected-operator))
        (equals (state (operator ~sop)) s))
  -->
      (spew t (preference :operator op1)
              (* *w1* (random-weight)))
  )

  (p@ lh-executive prefer-legal-ar2 ((s state)
                                     (op operator))
      (equal s (state op))
      (*no ((~op operator))
        (not-equal ~op op)
        (equal s (state ~op)))
      (*no ((~s state))
        (more-recent-state-p ~s s))
      (*no ((~pr preference 0.95))
        (equal (state (operator ~pr)) s))
      (*no ((~sop selected-operator))
        (equal (state (operator ~sop)) s))
  -->
      (spew t (preference :operator op)
              (* *w1* (random-weight)))
  )

  ))
((and (eq *accrual-scheme* 'multiplicative)
      (eq *preference-scheme* 'relative))
 (if *top-moves*

  (p@ lh-executive prefer-legal-mr1 ((s state)
                                     (op1 direct-operator)
                                     (op2 direct-operator))
      (equals s (state op1) (state op2))
      (legal-operator-p (contents s) op1)
      (*whole (not (legal-operator-p (contents s) op2)))
      (*no ((~s state))
        (more-recent-state-p ~s s))
      (*no ((~pr preference 0.01))
        (equal (state (operator ~pr)) s))
      (*no ((~sop selected-operator))
        (equals (state (operator ~sop)) s))
  -->
      (spew t (preference :operator op1)
             (* *w1* (random-weight)))
  )

  (p@ lh-executive prefer-legal-mr2 ((s state)
                                     (op operator))
      (equal s (state op))
      (*no ((~op operator))
```

```
              (not-equal ~op op)
              (equal s (state ~op)))
            (*no ((~s state))
              (more-recent-state-p ~s s))
            (*no ((~pr preference 0.01))
              (equal (state (operator ~pr)) s))
            (*no ((~sop selected-operator))
              (equal (state (operator ~sop)) s))
        -->
            (spew t (preference :operator op)
                    (* *w1* (random-weight)))
        )

        ))
)


; Prefer hill-climbing operators.

(cond ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-hill-climbing-aa ((s state)
                                                 (es end-state)
                                                 (op direct-operator))
           (equal s (state op))
           (hill-climbing-operator-p op es)
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.95))
             (equal (state (operator ~pr)) s))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op)
                   (* *w2* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-hill-climbing-ma ((s state)
                                                 (es end-state)
                                                 (op direct-operator))
           (equal s (state op))
           (hill-climbing-operator-p op es)
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.01))
             (equal (operator ~pr) op))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op)
                   (* *w2* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'additive)
```

```
            (eq *preference-scheme* 'relative))

       (p@ lh-executive prefer-hill-climbing-ar ((s state)
                                                (es end-state)
                                                (op1 direct-operator)
                                                (op2 direct-operator))
           (equals s (state op1) (state op2))
           (hill-climbing-operator-p op1 es)
           (*whole (not (hill-climbing-operator-p op2 es)))
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.95))
             (equal (state (operator ~pr)) s))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op1)
                   (* *w2* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'relative))

       (p@ lh-executive prefer-hill-climbing-mr ((s state)
                                                (es end-state)
                                                (op1 direct-operator)
                                                (op2 direct-operator))
           (equals s (state op1) (state op2))
           (hill-climbing-operator-p op1 es)
           (*whole (not (hill-climbing-operator-p op2 es)))
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.01))
             (equal (state (operator ~pr)) s))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op1)
                   (* *w2* (random-weight)))
       )

      )
)


; Prefer the steepest hill-climbing operator.

(cond ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-steepest-hill-climbing-aa ((s state)
                                                         (es end-state)
                                                         (op direct-operator))
           (equal s (state op))
           (hill-climbing-operator-p op es)
           (*no ((~op operator))
             (not-equal ~op op)
             (equal s (state ~op))
             (hill-climbing-operator-p ~op es)
```

```
      (steeper-climbing-operator-p ~op op es))
    (*no ((~s state))
      (more-recent-state-p ~s s))
    (*no ((~pr preference 0.95))
      (equal (state (operator ~pr)) s))
    (*no ((~sop selected-operator))
      (equals (state (operator ~sop)) s))
 -->
    (spew t (preference :operator op)
            (* *w3* (random-weight)))
 )

 )
((and (eq *accrual-scheme* 'multiplicative)
      (eq *preference-scheme* 'absolute))

 (p@ lh-executive prefer-steepest-hill-climbing-ma ((s state)
                                                    (es end-state)
                                                    (op direct-operator))
    (equal s (state op))
    (hill-climbing-operator-p op es)
    (*no ((~op operator))
      (not-equal ~op op)
      (equal s (state ~op))
      (hill-climbing-operator-p ~op es)
      (steeper-climbing-operator-p ~op op es))
    (*no ((~s state))
      (more-recent-state-p ~s s))
    (*no ((~pr preference 0.01))
      (equal (operator ~pr) op))
    (*no ((~sop selected-operator))
      (equals (state (operator ~sop)) s))
 -->
    (spew t (preference :operator op)
            (* *w3* (random-weight)))
 )

 )
((and (eq *accrual-scheme* 'additive)
      (eq *preference-scheme* 'relative))

 (p@ lh-executive prefer-steeper-hill-climbing-ar ((s state)
                                                   (es end-state)
                                                   (op1 direct-operator)
                                                   (op2 direct-operator))
    (equals s (state op1) (state op2))
    (steeper-climbing-operator-p op1 op2 es)
    (*no ((~s state))
      (more-recent-state-p ~s s))
    (*no ((~pr preference 0.95))
      (equal (state (operator ~pr)) s))
    (*no ((~sop selected-operator))
      (equals (state (operator ~sop)) s))
 -->
    (spew t (preference :operator op1)
            (* *w3* (random-weight)))
 )

 )
((and (eq *accrual-scheme* 'multiplicative)
```

```
            (eq *preference-scheme* 'relative))

      (p@ lh-executive prefer-steeper-hill-climbing-mr ((s state)
                                                        (es end-state)
                                                        (op1 direct-operator)
                                                        (op2 direct-operator))
          (equals s (state op1) (state op2))
          (steeper-climbing-operator-p op1 op2 es)
          (*no ((~s state))
            (more-recent-state-p ~s s))
          (*no ((~pr preference 0.01))
            (equal (operator ~pr) op1))
          (*no ((~sop selected-operator))
            (equals (state (operator ~sop)) s))
      -->
          (spew t (preference :operator op1)
                  (* *w3* (random-weight)))
      )

      )
)
```

```
; Prefer goal-based operators.

(cond ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-goal-aa ((s state)
                                        (g unblock-goal)
                                        (op indirect-operator))
           (equal s (state op))
           (equal g (goal op))
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.95))
             (equal (operator ~pr) op))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op)
                   (* *w4* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-goal-ma ((s state)
                                        (g unblock-goal)
                                        (op indirect-operator))
           (equal s (state op))
           (equal g (goal op))
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.01))
             (equal (operator ~pr) op))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
```

```
          (spew t (preference :operator op)
                  (* *w4* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'relative))

       (p@ lh-executive prefer-goal-ar ((s state)
                                         (g unblock-goal)
                                         (iop indirect-operator)
                                         (dop direct-operator))
           (equals s (state iop) (state dop))
           (equal g (goal iop))
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.95))
             (equal (state (operator ~pr)) s))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator iop)
                   (* *w4* (random-weight)))
       )

       )
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'relative))

      (p@ lh-executive prefer-goal-mr ((s state)
                                        (g unblock-goal)
                                        (iop indirect-operator)
                                        (dop direct-operator))
          (equals s (state iop) (state dop))
          (equal g (goal iop))
          (*no ((~s state))
            (more-recent-state-p ~s s))
          (*no ((~pr preference 0.01))
            (equal (operator ~pr) iop))
          (*no ((~sop selected-operator))
            (equals (state (operator ~sop)) s))
      -->
          (spew t (preference :operator iop)
                  (* *w4* (random-weight)))
      )

      )
)


; Prefer top-goal-based operators.

(cond ((and (eq *accrual-scheme* 'additive)
            (eq *preference-scheme* 'absolute))

       (p@ lh-executive prefer-top-goal-aa ((s state)
                                             (g unblock-goal)
                                             (op indirect-operator))
           (equal s (state op))
           (equal g (goal op))
```

```
     (*no ((~g goal))
       (more-recent-goal-p ~g g))
     (*no ((~s state))
       (more-recent-state-p ~s s))
     (*no ((~pr preference 0.95))
       (equal (operator ~pr) op))
     (*no ((~sop selected-operator))
       (equals (state (operator ~sop)) s))
 -->
     (spew t (preference :operator op)
             (* *w5* (random-weight)))
 )

 )
((and (eq *accrual-scheme* 'multiplicative)
      (eq *preference-scheme* 'absolute))

 (p@ lh-executive prefer-top-goal-ma ((s state)
                                      (g unblock-goal)
                                      (op indirect-operator))
     (equal s (state op))
     (equal g (goal op))
     (*no ((~g goal))
       (more-recent-goal-p ~g g))
     (*no ((~s state))
       (more-recent-state-p ~s s))
     (*no ((~pr preference 0.01))
       (equal (operator ~pr) op))
     (*no ((~sop selected-operator))
       (equals (state (operator ~sop)) s))
 -->
     (spew t (preference :operator op)
             (* *w5* (random-weight)))
 )

 )
((and (eq *accrual-scheme* 'additive)
      (eq *preference-scheme* 'relative))

 (p@ lh-executive prefer-top-goal-ar ((s state)
                                      (g1 unblock-goal)
                                      (g2 unblock-goal)
                                      (op1 indirect-operator)
                                      (op2 indirect-operator))
     (equals s (state op1) (state op2))
     (equal g1 (goal op1))
     (equal g2 (goal op2))
     (more-recent-goal-p g1 g2)
     (*no ((~s state))
       (more-recent-state-p ~s s))
     (*no ((~pr preference 0.95))
       (equal (state (operator ~pr)) s))
     (*no ((~sop selected-operator))
       (equals (state (operator ~sop)) s))
 -->
     (spew t (preference :operator op1)
             (* *w5* (random-weight)))
 )

 )
```

```
      ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'relative))

       (p@ lh-executive prefer-top-goal-mr ((s state)
                                            (g1 unblock-goal)
                                            (g2 unblock-goal)
                                            (op1 indirect-operator)
                                            (op2 indirect-operator))
           (equals s (state op1) (state op2))
           (equal g1 (goal op1))
           (equal g2 (goal op2))
           (more-recent-goal-p g1 g2)
           (*no ((~s state))
             (more-recent-state-p ~s s))
           (*no ((~pr preference 0.01))
             (equal (state (operator ~pr)) s))
           (*no ((~sop selected-operator))
             (equals (state (operator ~sop)) s))
       -->
           (spew t (preference :operator op1)
                   (* *w5* (random-weight)))
       )

      )
)


; For the chunking model only, prefer top-goal-based operators.

(when (and (eq *accrual-scheme* 'multiplicative)
           (eq *preference-scheme* 'absolute)
           *chunking*)

  (p@ lh-executive prefer-top-chunked-goal-mac ((s state)
                                                (g unblock-goal)
                                                (op chunked-indirect-move))
      (equal s (state op))
      (equal g (goal op))
      (*no ((~g goal))
        (more-recent-goal-p ~g g))
      (*no ((~s state))
        (more-recent-state-p ~s s))
      (*no ((~pr preference 0.01))
        (equal (operator ~pr) op))
      (*no ((~sop selected-operator))
        (equals (state (operator ~sop)) s))
      -->
      (spew t (preference :operator op)
              *weight*)
   )

)


;; When the *ACCRUAL-SCHEME* is 'MULTIPLICATIVE, must iteratively activate preferences
;; once they've been assigned their initial activations by the heuristic productions.

(cond ((and (eq *accrual-scheme* 'multiplicative)
            (eq *preference-scheme* 'absolute))
```

```
    (p@ lh-executive iteratively-activate-ma ((s state)
                                              (op operator)
                                              (pr preference .01))
        (equal s (state op))
        (equal (operator pr) op)
        (*no ((~s state))
          (more-recent-state-p ~s s))
        (*no ((~pr preference 0.95))
          (equal (state (operator ~pr)) s))
        (*no ((~sop selected-operator))
          (equals (state (operator ~sop)) s))
    -->
        (spew pr pr *w6*)
    )

    )
   ((and (eq *accrual-scheme* 'multiplicative)
        (eq *preference-scheme* 'relative))

    (p@ lh-executive iteratively-activate-mr ((s state)
                                              (op operator)
                                              (pr preference .01))
        (equal s (state op))
        (equal (operator pr) op)
        (*no ((~s state))
          (more-recent-state-p ~s s))
        (*no ((~pr preference 0.95))
          (equal (state (operator ~pr)) s))
        (*no ((~sop selected-operator))
          (equals (state (operator ~sop)) s))
    -->
        (spew pr pr *w6*)
    )

    )
)


;; Suppress preferences after a preferred operator has been selected.

(p@ lh-executive suppress-preference ((sop selected-operator)
                                      (pr preference 0.001))
    (equal (state (operator sop)) (state (operator pr)))
    (*no ((~sop selected-operator))
      (> (id ~sop) (id sop)))
-->
    (spew t pr (- *weight*))
)


;; Select the preferred operator.

;;; Select the operator whose preference first exceeds the activation threshold.
(p@ lh-executive select-among-preferences ((s state)
                                           (es end-state)
                                           (op operator)
                                           (pr preference 0.95))
    (equal s (state op))
    (equal (operator pr) op)
    (*no ((~s state))
```

```
      (more-recent-state-p ~s s))
    (*no ((~pr preference 0.95))
      (not-equal ~pr pr)
      (equal s (state (operator ~pr)))
      (subtypep (class-of (operator ~pr)) (find-class 'direct-operator))
      (subtypep (class-of op) (find-class 'direct-operator))
      (hill-climbing-operator-p (operator ~pr) es)
      (not (hill-climbing-operator-p op es)))
    (*no ((~pr preference 0.95))
      (not-equal ~pr pr)
      (equal s (state (operator ~pr)))
      (subtypep (class-of (operator ~pr)) (find-class 'direct-operator))
      (subtypep (class-of op) (find-class 'direct-operator))
      (steeper-climbing-operator-p (operator ~pr) op es))
    (*no ((~pr preference 0.95))
      (not-equal ~pr pr)
      (equal s (state (operator ~pr)))
      (subtypep (class-of (operator ~pr)) (find-class 'indirect-operator))
      (subtypep (class-of op) (find-class 'direct-operator)))
    (*no ((~pr preference 0.95))
      (not-equal ~pr pr)
      (equal s (state (operator ~pr)))
      (subtypep (class-of (operator ~pr)) (find-class 'indirect-operator))
      (subtypep (class-of op) (find-class 'indirect-operator))
      (more-recent-goal-p (goal (operator ~pr)) (goal op)))
    (*no ((~sop selected-operator))
      (equal (state (operator ~sop)) s))
-->
    (spew t (selected-operator :operator op)
            *weight*)
)


;; Suppress the preferred operator marker...

;;; ...after it has been applied.
(p@ lh-executive suppress-applied-selected-operator-marker ((sop selected-operator)
                                                            (op operator)
                                                            (bs state)
                                                            (as state))
    (equal (operator sop) op)
    (equal (state op) bs)
    (more-recent-state-p as bs)
    (contents-equal (perform-operator op bs) (contents as))
-->
    (spew t sop (- *weight*))
)

;;; ...if it cannot be applied because it is illegal.
(p@ lh-executive suppress-illegal-selected-operator-marker ((sop selected-operator)
                                                              (op operator)
                                                              (s state))
    (equal (operator sop) op)
    (equal (state op) s)
    (*whole (not (legal-operator-p (contents s) op)))
-->
    (spew t sop (- *weight*))
)

;;; ...if it cannot be applied, and has therefore caused an impasse, and has
```

```
;;; therefore led to the activation of a goal.
(p@ lh-executive suppress-unapplied-selected-operator-marker ((g goal)
                                                              (sop selected-operator)
                                                              (op operator))
    (equals (operator g) (operator sop) op)
-->
    (spew t sop (- *weight*))
)



;;;
;;; TOH model productions.
;;;




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (5) The RH-Executive Center.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal model productions.
;;;


;; Suppress goals.

;;; When the most recent goal G is satisfied by the current state S, then
;;; suppress the completed goal.
(p@ rh-executive suppress-satisfied-goal ((g goal)
                                          (s state))
    (satisfied-p g s)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~s state))
      (more-recent-state-p ~s s))
-->
    (spew t g (- *weight*))
)



;;;
;;; TOH model productions.
;;;


;; Propose goals.

(cond

((not *chunking*)

(p@ rh-executive propose-unblock-disk-goal ((p puzzle)
```

```
                                            (ep end-puzzle)
                                            (m move-mi)
                                            (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (blocked-disk-p (contents p) (disk m))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (occupied-position-p (contents p) (dest-peg m) (dest-pos m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (> (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~udg unblock-disk-goal))
      (equal (operator ~udg) m)
      (equal (disk ~udg) (disk m))
      (equal (source-peg ~udg) (source-peg m))
      (equal (dest-peg ~udg) (dest-peg m))
      (equal (dest-pos ~udg) (dest-pos m)))
-->
    (spew t (unblock-disk-goal :operator m
                               :disk (disk m)
                               :source-peg (source-peg m)
                               :dest-peg (dest-peg m)
                               :dest-pos (dest-pos m))
            *weight*)

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Subgoal to unblock DISK~A on ~A. (~A)" (disk m) (source-peg m) *cycles*))
)

(p@ rh-executive propose-unblock-position-goal ((p puzzle)
                                                (ep end-puzzle)
                                                (m move-mi)
                                                (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (occupied-position-p (contents p) (dest-peg m) (dest-pos m))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (blocked-disk-p (contents p) (disk m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (< (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~upg unblock-position-goal))
      (equal (operator ~upg) m)
      (equal (source-peg ~upg) (source-peg m))
      (equal (dest-peg ~upg) (dest-peg m))
      (equal (dest-pos ~upg) (dest-pos m)))
-->
    (spew t (unblock-position-goal :operator m
                                   :source-peg (source-peg m)
                                   :dest-peg (dest-peg m)
                                   :dest-pos (dest-pos m))
            *weight*)

    (when *toh-tracing*
```

```
      (format t "~&")
      (goal-indent)
      (format t "Subgoal to unblock position ~A on ~A. (~A)" (dest-pos m) (dest-peg m) *cycles*))
)

)

(t

(p@ rh-executive propose-unblock-disk-goal ((p puzzle)
                                            (ep end-puzzle)
                                            (m move-mi)
                                            (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (blocked-disk-p (contents p) (disk m))
    (*whole (not (and (not-solved-p p ep)
                      (> (largest-out-of-place-disk (contents p) (contents ep)) 3)
                      (= (disk m) 3)
                      (on-top-of-p (contents p) 2 3)
                      (on-top-of-p (contents p) 1 2)
                      (top-disk-p (contents p) 1))))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (occupied-position-p (contents p) (dest-peg m) (dest-pos m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (> (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~udg unblock-disk-goal))
      (equal (operator ~udg) m)
      (equal (disk ~udg) (disk m))
      (equal (source-peg ~udg) (source-peg m))
      (equal (dest-peg ~udg) (dest-peg m))
      (equal (dest-pos ~udg) (dest-pos m)))
-->
    (spew t (unblock-disk-goal :operator m
                               :disk (disk m)
                               :source-peg (source-peg m)
                               :dest-peg (dest-peg m)
                               :dest-pos (dest-pos m))
            *weight*)

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Subgoal to unblock DISK~A on ~A. (~A)" (disk m) (source-peg m) *cycles*))
)

(p@ rh-executive propose-unblock-position-goal ((p puzzle)
                                                (ep end-puzzle)
                                                (m move-mi)
                                                (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (occupied-position-p (contents p) (dest-peg m) (dest-pos m))
    (*whole (not (and (not-solved-p p ep)
                      (> (largest-out-of-place-disk (contents p) (contents ep)) 2)
                      (equal (peg-of (contents p) 2) (dest-peg m))
                      (= (position-of (contents p) 2) (dest-pos m))
```

```
                      (on-top-of-p (contents p) 1 2)
                      (top-disk-p (contents p) 1))))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (blocked-disk-p (contents p) (disk m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (< (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~upg unblock-position-goal))
      (equal (operator ~upg) m)
      (equal (source-peg ~upg) (source-peg m))
      (equal (dest-peg ~upg) (dest-peg m))
      (equal (dest-pos ~upg) (dest-pos m)))
-->
    (spew t (unblock-position-goal :operator m
                                   :source-peg (source-peg m)
                                   :dest-peg (dest-peg m)
                                   :dest-pos (dest-pos m))
            *weight*)

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Subgoal to unblock position ~A on ~A. (~A)" (dest-pos m) (dest-peg m) *cycles*))
)

(p@ rh-executive propose-chunked-unblock-disk-goal ((p puzzle)
                                                    (ep end-puzzle)
                                                    (m move-mi)
                                                    (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (blocked-disk-p (contents p) (disk m))
    (not-solved-p p ep)
    (> (largest-out-of-place-disk (contents p) (contents ep)) 3)
    (= (disk m) 3)
    (on-top-of-p (contents p) 2 3)
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (occupied-position-p (contents p) (dest-peg m) (dest-pos m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (> (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~udg unblock-disk-goal))
      (equal (operator ~udg) m)
      (equal (disk ~udg) (disk m))
      (equal (source-peg ~udg) (source-peg m))
      (equal (dest-peg ~udg) (dest-peg m))
      (equal (dest-pos ~udg) (dest-pos m)))
-->
    (spew t (chunked-unblock-disk-goal :operator m
                                       :disk (disk m)
                                       :source-peg (source-peg m)
                                       :dest-peg (dest-peg m)
                                       :dest-pos (dest-pos m))
            *weight*)
```

```
    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "CHUNKED Subgoal to unblock DISK~A on ~A. (~A)" (disk m) (source-peg m)
*cycles*))
)

(p@ rh-executive propose-chunked-unblock-position-goal ((p puzzle)
                                                        (ep end-puzzle)
                                                        (m move-mi)
                                                        (sop selected-operator))
    (equal (state m) p)
    (equal (operator sop) m)
    (not-solved-p p ep)
    (> (largest-out-of-place-disk (contents p) (contents ep)) 2)
    (equal (peg-of (contents p) 2) (dest-peg m))
    (= (position-of (contents p) 2) (dest-pos m))
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~d disk))
      (blocked-disk-p (contents p) (disk m))
      (equal (peg-of (contents p) (disk ~d)) (dest-peg m))
      (equal (position-of (contents p) (disk ~d)) (dest-pos m))
      (< (disk ~d) (largest-source-blocking-disk (contents p) (disk m))))
    (*no ((~upg unblock-position-goal))
      (equal (operator ~upg) m)
      (equal (source-peg ~upg) (source-peg m))
      (equal (dest-peg ~upg) (dest-peg m))
      (equal (dest-pos ~upg) (dest-pos m)))
-->
    (spew t (chunked-unblock-position-goal :operator m
                                           :source-peg (source-peg m)
                                           :dest-peg (dest-peg m)
                                           :dest-pos (dest-pos m))
            *weight*)

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "CHUNKED Subgoal to unblock position ~A on ~A. (~A)" (dest-pos m) (dest-peg m)
*cycles*))
)

))



;;

(p@ rh-executive suppress-superfluous-unblock-disk-goal ((g1 unblock-disk-goal)
                                                         (g2 unblock-disk-goal))
    (equal (disk g1) (disk g2))
    (equal (source-peg g1) (source-peg g2))
    (equal (dest-peg g1) (dest-peg g2))
    (equal (dest-pos g1) (dest-pos g2))
    (more-recent-goal-p g1 g2)
-->
```

```
    (spew t g2 (- *weight*))

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Suppress superfluous unblock-disk-goal. (~A)" *cycles*))
)

(p@ rh-executive suppress-superfluous-unblock-position-goal ((g1 unblock-position-goal)
                                                              (g2 unblock-position-goal))
    (equal (source-peg g1) (source-peg g2))
    (equal (dest-peg g1) (dest-peg g2))
    (equal (dest-pos g1) (dest-pos g2))
    (more-recent-goal-p g1 g2)
-->
    (spew t g2 (- *weight*))

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Suppress superfluous unblock-position-goal. (~A)" *cycles*))
)


;; Propose the indirect moves that satisfy outstanding goals.

(cond

(*top-goal-moves-only*

(p@ rh-executive propose-unblock-disk-move-top ((p puzzle)
                                                 (g unblock-disk-goal)
                                                 (d disk))
    (blocked-disk-p (contents p) (disk g))
    (on-top-of-p (contents p) (disk d) (disk g))
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) (disk d))
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (buffer-peg-position (contents p) (buffer-peg g) (disk d))))
-->
    (spew t (indirect-move :goal g
                           :state p
                           :disk (disk d)
                           :source-peg (source-peg g)
                           :dest-peg (buffer-peg g)
                           :dest-pos (buffer-peg-position (contents p) (buffer-peg g) (disk d)))
            *weight*)
)

(p@ rh-executive propose-unblock-position-move-top ((p puzzle)
                                                     (g unblock-position-goal)
                                                     (d disk))
    (occupied-position-p (contents p) (dest-peg g) (dest-pos g))
```

```
    (equal (peg-of (contents p) (disk d)) (dest-peg g))
    (equal (position-of (contents p) (disk d)) (dest-pos g))
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) (disk d))
      (equal (source-peg ~im) (dest-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (buffer-peg-position (contents p) (buffer-peg g) (disk d))))
-->
    (spew t (indirect-move :goal g
                           :state p
                           :disk (disk d)
                           :source-peg (dest-peg g)
                           :dest-peg (buffer-peg g)
                           :dest-pos (buffer-peg-position (contents p) (buffer-peg g) (disk d)))
            *weight*)
)

)

(t

(p@ rh-executive propose-unblock-disk-move-all ((p puzzle)
                                                (g unblock-disk-goal)
                                                (d disk))
    (blocked-disk-p (contents p) (disk g))
    (on-top-of-p (contents p) (disk d) (disk g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) (disk d))
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (buffer-peg-position (contents p) (buffer-peg g) (disk d))))
-->
    (spew t (indirect-move :goal g
                           :state p
                           :disk (disk d)
                           :source-peg (source-peg g)
                           :dest-peg (buffer-peg g)
                           :dest-pos (buffer-peg-position (contents p) (buffer-peg g) (disk d)))
            *weight*)
)

(p@ rh-executive propose-unblock-position-move-all ((p puzzle)
                                                    (g unblock-position-goal)
                                                    (d disk))
    (occupied-position-p (contents p) (dest-peg g) (dest-pos g))
    (equal (peg-of (contents p) (disk d)) (dest-peg g))
    (equal (position-of (contents p) (disk d)) (dest-pos g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
```

```
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) (disk d))
      (equal (source-peg ~im) (dest-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (buffer-peg-position (contents p) (buffer-peg g) (disk d))))
-->
    (spew t (indirect-move :goal g
                           :state p
                           :disk (disk d)
                           :source-peg (dest-peg g)
                           :dest-peg (buffer-peg g)
                           :dest-pos (buffer-peg-position (contents p) (buffer-peg g) (disk d)))
            *weight*)
)

))

(when *chunking*

; The next three productions are presumably the result of a learning (chunking)
; process.  They propose the indirect moves that solve embedded 2-disk tower-to-
; tower problems without the need for goals.

(p@ rh-executive propose-unblock-position-move-top-only-2-1 ((p puzzle)
                                                             (g chunked-unblock-position-goal)
                                                             (d disk))
    (equal (peg-of (contents p) 2) (dest-peg g))
    (= (position-of (contents p) 2) (dest-pos g))
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
      (equal (source-peg ~im) (dest-peg g))
      (equal (dest-peg ~im) (source-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (source-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (dest-peg g)
                                   :dest-peg (source-peg g)
                                   :dest-pos (top-empty-position (contents p) (source-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-position-move-top-only-2-2 ((p puzzle)
                                                             (g chunked-unblock-position-goal)
                                                             (d disk))
    (equal (peg-of (contents p) 2) (dest-peg g))
    (top-disk-p (contents p) 2)
    (equal (peg-of (contents p) 1) (source-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
```

```
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 2)
      (equal (source-peg ~im) (dest-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (buffer-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 2
                                   :source-peg (dest-peg g)
                                   :dest-peg (buffer-peg g)
                                   :dest-pos (top-empty-position (contents p) (buffer-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-position-move-top-only-2-3 ((p puzzle)
                                                             (g chunked-unblock-position-goal)
                                                             (d disk))
    (equal (peg-of (contents p) 2) (buffer-peg g))
    (top-disk-p (contents p) 2)
    (equal (peg-of (contents p) 1) (source-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (buffer-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (source-peg g)
                                   :dest-peg (buffer-peg g)
                                   :dest-pos (top-empty-position (contents p) (buffer-peg g)))
            *weight*)
)

; The next eight productions are presumably the result of a learning (chunking)
; process.  They propose the indirect moves that solve embedded 3-disk tower-to-
; tower problems without the need for goals.

(p@ rh-executive propose-unblock-disk-move-top-only-3-1 ((p puzzle)
                                                         (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (source-peg g))
    (on-top-of-p (contents p) 2 3)
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
```

```
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (dest-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (dest-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (source-peg g)
                                   :dest-peg (dest-peg g)
                                   :dest-pos (top-empty-position (contents p) (dest-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-2 ((p puzzle)
                                                         (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (source-peg g))
    (on-top-of-p (contents p) 2 3)
    (equal (peg-of (contents p) 1) (dest-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 2)
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (buffer-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 2
                                   :source-peg (source-peg g)
                                   :dest-peg (buffer-peg g)
                                   :dest-pos (top-empty-position (contents p) (buffer-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-3 ((p puzzle)
                                                         (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (source-peg g))
    (equal (peg-of (contents p) 2) (buffer-peg g))
    (top-disk-p (contents p) 2)
    (equal (peg-of (contents p) 1) (dest-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
```

```
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
      (equal (source-peg ~im) (dest-peg g))
      (equal (dest-peg ~im) (buffer-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (buffer-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (dest-peg g)
                                   :dest-peg (buffer-peg g)
                                   :dest-pos (top-empty-position (contents p) (buffer-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-4 ((p puzzle)
                                                        (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (source-peg g))
    (equal (peg-of (contents p) 2) (buffer-peg g))
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 3)
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (dest-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (dest-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 3
                                   :source-peg (source-peg g)
                                   :dest-peg (dest-peg g)
                                   :dest-pos (top-empty-position (contents p) (dest-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-5 ((p puzzle)
                                                        (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (dest-peg g))
    (equal (peg-of (contents p) 2) (buffer-peg g))
    (on-top-of-p (contents p) 1 2)
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
```

```
      (equal (source-peg ~im) (buffer-peg g))
      (equal (dest-peg ~im) (source-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (source-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (buffer-peg g)
                                   :dest-peg (source-peg g)
                                   :dest-pos (top-empty-position (contents p) (source-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-6 ((p puzzle)
                                                         (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (dest-peg g))
    (equal (peg-of (contents p) 2) (buffer-peg g))
    (top-disk-p (contents p) 2)
    (equal (peg-of (contents p) 1) (source-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 2)
      (equal (source-peg ~im) (buffer-peg g))
      (equal (dest-peg ~im) (dest-peg g))
      (equal (dest-pos ~im) (top-empty-position (contents p) (dest-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 2
                                   :source-peg (buffer-peg g)
                                   :dest-peg (dest-peg g)
                                   :dest-pos (top-empty-position (contents p) (dest-peg g)))
            *weight*)
)

(p@ rh-executive propose-unblock-disk-move-top-only-3-7 ((p puzzle)
                                                         (g chunked-unblock-disk-goal))
    (= (disk g) 3)
    (equal (peg-of (contents p) 3) (dest-peg g))
    (on-top-of-p (contents p) 2 3)
    (top-disk-p (contents p) 2)
    (equal (peg-of (contents p) 1) (source-peg g))
    (top-disk-p (contents p) 1)
    (*no ((~g goal))
      (more-recent-goal-p ~g g))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~im indirect-move))
      (equal (goal ~im) g)
      (equal (state ~im) p)
      (equal (disk ~im) 1)
      (equal (source-peg ~im) (source-peg g))
      (equal (dest-peg ~im) (dest-peg g))
```

```
      (equal (dest-pos ~im) (top-empty-position (contents p) (dest-peg g))))
-->
    (spew t (chunked-indirect-move :goal g
                                   :state p
                                   :disk 1
                                   :source-peg (source-peg g)
                                   :dest-peg (dest-peg g)
                                   :dest-pos (top-empty-position (contents p) (dest-peg g)))
            *weight*)
)

)



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (6) The LH-Spatial Center.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal model productions.
;;;


;; Suppress old states.

(case *suppress-old-states*

(all

(p@ lh-spatial suppress-old-state ((news state)
                                   (olds state))
    (more-recent-state-p news olds)
-->
    (spew t olds (- *weight*))
)

)

(non-goal

(p@ lh-spatial suppress-non-goal-state ((news state)
                                        (olds state))
    (more-recent-state-p news olds)
    (*no ((~g goal))
      (equal (state (operator ~g)) olds))
-->
    (spew t olds (- *weight*))
)

))



;;;
```

```
;;; TOH model productions.
;;;


;; Perform moves.

(p@ lh-spatial perform-move ((sop selected-operator)
                             (m move-mi)
                             (p puzzle))
    (equal (operator sop) m)
    (equal (state m) p)
    (legal-operator-p (contents p) m)
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
-->
    (spew t (puzzle :contents (perform-operator m p))
            *weight*)

    (when *toh-tracing*
      (format t "~&")
      (goal-indent)
      (format t "Move DISK~A from ~A to ~A. (~A)" (disk m) (source-peg m) (dest-peg m) *cycles*))

    (when *record-moves-p*
      (setq *move-record* (nconc *move-record* (list *cycles*))))

    (when *constrained*
      (let* ((move (first *constrained-moves*))
             (disk (second move))
             (dest-peg (third move))
             (error-p (not (and (= disk (disk m))
                                (eq dest-peg (dest-peg m))))))
        (case (fourth move)
          (0
           (incf *tot0*)
           (when error-p
             (incf *err0*)))
          (1
           (incf *tot1*)
           (when error-p
             (incf *err1*)))
          (t
           (incf *tot2+*)
           (when error-p
             (incf *err2+*))))))
)



;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (7) The RH-Spatial Center.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Fronto-Parietal model productions.
```

```
;;;

;; Suppress an operators if...

;;; ...it was not selected.
(p@ rh-spatial suppress-unselected-operator ((sop selected-operator)
                                             (op1 operator)
                                             (op2 operator))
    (equal (operator sop) op1)
    (not-equal op1 op2)
    (equal (state op1) (state op2))
-->
    (spew t op2 (- *weight*))
)

;;; ...it was selected and successfully applied.
(p@ rh-spatial suppress-applied-selected-operator ((sop selected-operator)
                                                     (op operator)
                                                     (bs state)
                                                     (as state))
    (equal (operator sop) op)
    (equal (state op) bs)
    (more-recent-state-p as bs)
    (contents-equal (perform-operator op bs) (contents as))
-->
    (spew t op (- *weight*))

    (when *constrained*
      (pop *constrained-moves*))
)

;;; ...it cannot be applied because it is illegal.
(p@ rh-spatial suppress-illegal-selected-operator ((sop selected-operator)
                                                   (op operator)
                                                   (s state))
    (equal (operator sop) op)
    (equal (state op) s)
    (*whole (not (legal-operator-p (contents s) op)))
-->
    (spew t op (- *weight*))
)

;;; ...if it cannot be applied, and has therefore caused an impasse, and has
;;; therefore led to the activation of a goal.
(p@ rh-spatial suppress-unapplied-selected-operator ((g goal)
                                                     (sop selected-operator)
                                                     (op operator))
    (equals (operator g) (operator sop) op)
-->
    (spew t op (- *weight*))
)



;;;
;;; TOH model productions.
;;;
```

```
;; Propose direct moves, which are perceptually-triggered.  They strive to
;; increase the similarity between the current and ending configurations.
;; No attention is paid as to whether the preconditions of these moves are
;; satisfied; this is the concern of other centers.

;;; Proposes direct moves that place out-of-place disks at their peg position
;;; in the ending configuration.
(p@ rh-spatial propose-ending-move ((g solve-puzzle-goal)
                                    (p puzzle)
                                    (ep end-puzzle)
                                    (d disk))
    (id g)
    (not-solved-p p ep)
    (not-equal (peg-of (contents p) (disk d)) (peg-of (contents ep) (disk d)))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~dm direct-move))
      (equal (state ~dm) p)
      (equal (disk ~dm) (disk d))
      (equal (source-peg ~dm) (peg-of (contents p) (disk d)))
      (equal (dest-peg ~dm) (peg-of (contents ep) (disk d)))
      (equal (dest-pos ~dm) (position-of (contents ep) (disk d))))
-->
    (spew t (direct-move :state p
                         :disk (disk d)
                         :source-peg (peg-of (contents p) (disk d))
                         :dest-peg (peg-of (contents ep) (disk d))
                         :dest-pos (position-of (contents ep) (disk d)))
            *weight*)
)

(when *top-moves*

;;; Propose perceptually-available or salient direct moves that place disks
;;; that are on top of pegs on top of other pegs.
(p@ rh-spatial propose-top-move ((g solve-puzzle-goal)
                                 (p puzzle)
                                 (ep end-puzzle)
                                 (d disk)
                                 (dp peg))
    (id g)
    (not-solved-p p ep)
    (top-disk-p (contents p) (disk d))
    (not-equal (peg-of (contents p) (disk d)) (peg dp))
    (*no ((~p puzzle))
      (more-recent-state-p ~p p))
    (*no ((~dm direct-move))
      (equal (state ~dm) p)
      (equal (disk ~dm) (disk d))
      (equal (source-peg ~dm) (peg-of (contents p) (disk d)))
      (equal (dest-peg ~dm) (peg dp))
      (equal (dest-pos ~dm) (top-empty-position (contents p) (peg dp))))
-->
    (spew t (direct-move :state p
                         :disk (disk d)
                         :source-peg (peg-of (contents p) (disk d))
                         :dest-peg (peg dp)
                         :dest-pos (top-empty-position (contents p) (peg dp)))
            *weight*)
)
```

```
)




;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;
;;;; (8) Support Code.
;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;



;;;
;;; Top-level commands that automate simulations of TOH problem solving.
;;;


;; The default value for the maximum number of cycles simulations run before
;; the model gives up.

(defvar *max-cycs* 2000)


;; Top-level command to run a simulation.  Takes three obligatory arguments:
;;      disks:  number of disks of the problem          3
;;      start:  starting configuration of the problem  '((3 2 1) () ())
;;        end:  ending configuration of the problem    '(() () (3 2 1)))
;; and one optional argument:
;;      max-cycs: maximum number of cycles to run the simulation before giving
;;                up (defaults to the value of *MAX-CYCS*)

(defmacro sim (&rest args)
  `(impl-sim ,@args))

(defun impl-sim (disks start end &key (max-cycs *max-cycs*))
  (reset)
  (when *record-moves-p*
    (setq *move-record* ()))
  (spew t (solve-puzzle-goal)
        *weight*)
  (spew t (puzzle :contents (make-instance 'configuration
                              :disks disks
                              :peg1 (first start)
                              :peg2 (second start)
                              :peg3 (third start)))
        *weight*)
  (spew t (end-puzzle :contents (make-instance 'configuration
                                  :disks disks
                                  :peg1 (first end)
                                  :peg2 (second end)
                                  :peg3 (third end)))
        *weight*)
  (do ((n 1 (1+ n)))
      ((> n disks))
    (spew t (disk :disk n)
          *weight*))
  (spew t (peg :peg 'peg1)
        *weight*)
```

```
  (spew t (peg :peg 'peg2)
          *weight*)
  (spew t (peg :peg 'peg3)
          *weight*)
  (run max-cycs))


;; Top-level command to print a summary of a simulation run, including the
;; moves made, the capacity utilizations of the centers, and the final contents
;; of declarative memory.

(defun summ ()
  (when *record-moves-p*
    (format t "~&~%NUM~ADELTA~ATOTAL" #\tab #\tab)
    (do* ((num 1 (1+ num))
          (prev-cycs 0 (first cycs))
          (cycs *move-record* (rest cycs)))
         ((null cycs))
      (let* ((first-cycs (first cycs))
             (delta-cycs (- first-cycs prev-cycs)))
      (format t "~%~D~A~D~A~D" num #\tab delta-cycs #\tab first-cycs))))
  (format t "~%")
  (history@ (lh-executive rh-executive lh-spatial rh-spatial)
            :combination avg
            :measure act)
  (format t "~&~%")
  (dm base-state
      operator
      preference
      selected-operator
      base-goal))


;; Test functions for simulating the solution of standard 3-disk, 4-disk, and
;; 5-disk tower-to-tower problems.

(defun test3 ()
  (reset)
  (sim 3 '((3 2 1) () ()      )
         '(()      () (3 2 1)))
  (summ)
  (values))

(defun test4 ()
  (reset)
  (sim 4 '((4 3 2 1) () ()        )
         '(()        () (4 3 2 1)))
  (summ)
  (values))

(defun test5 ()
  (reset)
  (sim 5 '((5 4 3 2 1) () ()          )
         '(()          () (5 4 3 2 1)))
  (summ)
  (values))
```

APPENDIX B: ANNOTATED SIMULATION TRACE

This appendix contains a trace of the TOH model solving the standard 2-disk tower-to-tower problem. Recall that there are 48 variants of the model arising from the possible combinations of the four binary-valued design decisions and one ternary valued design decision. The source code listed in Appendix A emphasized the one where the *PREFERENCE-SCHEME*, *ACCRUAL-SCHEME*, *TOP-MOVES*, *TOP-GOAL-MOVES-ONLY*, *SUPPRESS-OLD-STATES* variables take the values ABSOLUTE, MULTIPLICATIVE, YES, NO, and ALL, respectively. It is this model variant that produced this trace, which is annotated to aid the reader.

The simulation is initiated with a top-level command that takes three parameters: the number of disks, the starting configuration, and the ending configuration. Smaller disks are denoted by smaller numbers. In the starting configuration of the standard 2-disk tower-to-tower problem, the small disk is on top of large disk, which is on the left peg. In the ending configuration, the small disk is again on top of the large disk, which is on the right peg.

```
? (set-tracing-p   t)
? (sim  2  '((2  1)  ()  ())
           '(()      ()  (2  1)))
```

The line of asterisks indicates the start of a new cycle – in this, case the first cycle of the simulation. Within each cycle, the activity of all centers is documented. Nothing should be read into the order in which centers are listed within a cycle; the control structure of 4CAPS is fully parallel.

```
*******************************************************************************
                                  CYCLE: 1
```

Recall that RH-Spatial is specialized for visual attention. In particular, its productions compare the current and ending configurations, which are either perceptually available in the external environment or visuospatial representations in LH-Spatial, and propose moves that increase the similarity between the two. In this case, two ending moves are proposed. Recall that an ending move directly transfers a disk that is out-of-place in the current configuration to its peg position in the ending configuration. One of the proposed ending moves transfers the large disk (2) from the left peg to the right peg; the other transfers the small disk from the left peg to the right peg. Because the value of the *TOP-MOVES* variable is YES, top moves are also proposed. Recall that a top move transfers a disk from the top of one peg to the top of another peg. One of the proposed top moves transfers the lone top disk, the small disk (1), from the left peg to the middle peg; the other transfers it from the left peg to the right peg.

```
===============================================================================
RH-SPATIAL: 1
-------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
-->
```

```
  SPEW:   0.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 8 :PEG PEG3)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 7 :PEG PEG2)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

This completes cycle 1. No other centers were active. This is relatively uncommon occurrence in 4CAPS models, as we will soon see. It is the case here only because this is the first move of the problem, and there are not yet many representations to trigger the firing of productions.

Turning to cycle 2, recall that LH-Executive is specialized for selecting between possible operators via preference-based reasoning. The particular dynamics of selection are given by the `*PREFERENCE-SCHEME*` and `*ACCRUAL-SCHEME*` variables, which in this model variant take the values `ABSOLUTE` and `MULTIPLICATIVE`, respectively. This means that preferences will initially be activated to a level commensurate with the selection heuristics their operators satisfy; each will be evaluated on its own merits, not relative to its competitors. This also means that on future cycles, the preferences will activate themselves iteratively until one exceeds threshold. Because there are no goals at the present time, the only heuristics that apply favor legal operators, hill-climbing operators, and the steepest hill-climbing operator.

```
****************************************************************************
                                  CYCLE: 2

============================================================================
LH-EXECUTIVE: 1
----------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS
0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
----------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
----------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
    With: 0.03
    Req:  0.03
    Cont: 0.03 (* 1.00 = 0.03)
----------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: 0.03
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
----------------------------------------------------------------------------
PREFER-STEEPEST-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: 0.05
```

```
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
********************************************************************************
```

Over the next few cycles, the preferences iteratively activate themselves.

```
********************************************************************************
                                   CYCLE: 3
================================================================================
LH-EXECUTIVE: 2
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.08:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   0.08:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: 0.15
    Req:  0.23
    Cont: 0.23 (* 1.00 = 0.23)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  0.03:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
-->
  SPEW:   0.03:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
    With: 0.05
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS
0)
  0.01:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
********************************************************************************

********************************************************************************
                                   CYCLE: 4
================================================================================
```

```
LH-EXECUTIVE: 3
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.23:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   0.23:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: 0.45
    Req:  0.68
    Cont: 0.68 (* 1.00 = 0.68)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  0.08:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
-->
  SPEW:   0.08:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
    With: 0.15
    Req:  0.23
    Cont: 0.23 (* 1.00 = 0.23)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS
0)
  0.04:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
********************************************************************************

********************************************************************************
                                   CYCLE: 5
================================================================================
LH-EXECUTIVE: 4
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.68:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   0.68:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: 1.35
```

```
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  0.23:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
-->
  SPEW:   0.23:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
    With: 0.45
    Req:  0.68
    Cont: 0.68 (* 1.00 = 0.68)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.11:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS
0)
  0.11:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
********************************************************************************
```

By cycle 6, a preference achieves an activation above threshold (0.95), and the corresponding move is selected. This is an ending move, the transfer of the large disk (2) from the left peg to the bottom position of the right peg, where it appears in the ending configuration.

```
********************************************************************************
                                   CYCLE: 6
================================================================================
LH-EXECUTIVE: 5
--------------------------------------------------------------------------------
SELECT-AMONG-PREFERENCES
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   0.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

However, the selected move cannot be performed because the disk to be moved, the large disk (2), is blocked from above by the small disk (1), and thus a precondition of disk movement is unsatisfied. Therefore, on cycle 7, a goal is created to satisfy this precondition by unblocking the large disk. The model prints a message announcing this occurrence.

```
Subgoal to unblock DISK2 on PEG1. (7)

********************************************************************************
                                   CYCLE: 7
```

The goal itself is created by a production in RH-Executive, which is specialized for goal-related functions.

```
================================================================================
RH-EXECUTIVE: 1
--------------------------------------------------------------------------------
PROPOSE-UNBLOCK-DISK-GOAL
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   0.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-
PEG PEG3 :DEST-POS 0 :DISK 2)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
```

Note that cycle 7 brings the first example of parallel information processing *across* centers. Productions also fire in LH-Executive to suppress the partial products – the preferences and selected operator marker – of the just-completed iteration of problem solving.

```
================================================================================
LH-EXECUTIVE: 6
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  0.34:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 17 :OPERATOR #<DIRECT-MOVE 9>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  0.34:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 16 :OPERATOR #<DIRECT-MOVE 10>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
```

```
  0.68:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
-->
  SPEW:   0.68:   (PREFERENCE :ID 15 :OPERATOR #<DIRECT-MOVE 11>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
-->
  SPEW:   1.00:   (PREFERENCE :ID 13 :OPERATOR #<DIRECT-MOVE 12>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-ILLEGAL-SELECTED-OPERATOR-MARKER
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
-->
  SPEW:   1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Productions also fire in RH-Spatial to suppress direct moves proposed during the just-completed iteration of problem solving, which are now "stale."

```
================================================================================
RH-SPATIAL: 2
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS
0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 9 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 10 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
```

```
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 11 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-ILLEGAL-SELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 18 :OPERATOR #<DIRECT-MOVE 12>)
  1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 12 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
********************************************************************************
```

A new iteration of problem solving begins on cycle 8. As before, four direct moves are proposed by productions in RH-Spatial. In addition, an indirect move is proposed by a production RH-Executive. This move is prompted by the new goal, and it proposes transferring the small disk, which is blocking movement of the large disk from the left peg to the right peg, to the middle peg, which serves as a temporary buffer. This move is indirect in that it does not increase the similarity between the current and ending configurations.

```
********************************************************************************
                                   CYCLE: 8
================================================================================
RH-EXECUTIVE: 2
--------------------------------------------------------------------------------
PROPOSE-UNBLOCK-DISK-MOVE-ALL
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0 :DISK 2)
  1.00:   (DISK :ID 4 :DISK 1)
-->
  SPEW:   0.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1
:SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
================================================================================
RH-SPATIAL: 3
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
-->
```

```
  SPEW:   0.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 8 :PEG PEG3)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 7 :PEG PEG2)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

The heuristic productions in LH-Executive fire, activating preferences for the five proposed moves. For the first time, heuristics apply that favor operators that satisfy goals and, in particular, operators that satisfy the topmost (i.e., most recent) goal on the stack.

```
********************************************************************************
                                   CYCLE: 9
================================================================================
LH-EXECUTIVE: 7
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-
POS 0)
-->
```

```
  SPEW:   0.00:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
    With: 0.03
    Req:  0.03
    Cont: 0.03 (* 1.00 = 0.03)
--------------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
    With: 0.03
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
--------------------------------------------------------------------------------
PREFER-STEEPEST-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
    With: 0.05
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
--------------------------------------------------------------------------------
PREFER-GOAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0 :DISK 2)
-->
  SPEW:   0.00:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
    With: 0.05
    Req:  0.15
    Cont: 0.15 (* 1.00 = 0.15)
--------------------------------------------------------------------------------
```

```
PREFER-TOP-GOAL-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0 :DISK 2)
-->
  SPEW:   0.00:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
    With: 0.10
    Req:  0.15
    Cont: 0.15 (* 1.00 = 0.15)
********************************************************************************
```

Over the next few cycles, preferences accrue activation.

```
********************************************************************************
                                   CYCLE: 10
================================================================================
LH-EXECUTIVE: 8
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  0.15:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
-->
  SPEW:   0.15:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
    With: 0.30
    Req:  0.45
    Cont: 0.45 (* 1.00 = 0.45)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.08:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
-->
  SPEW:   0.08:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
    With: 0.15
    Req:  0.23
    Cont: 0.23 (* 1.00 = 0.23)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  0.03:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
-->
  SPEW:   0.03:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
    With: 0.05
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
-->
```

```
  SPEW:   0.01:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
********************************************************************************

********************************************************************************
                                   CYCLE: 11
================================================================================
LH-EXECUTIVE: 9
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  0.45:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
-->
  SPEW:   0.45:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
    With: 0.90
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.23:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
-->
  SPEW:   0.23:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
    With: 0.45
    Req:  0.68
    Cont: 0.68 (* 1.00 = 0.68)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
  0.08:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
-->
  SPEW:   0.08:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
    With: 0.15
    Req:  0.23
    Cont: 0.23 (* 1.00 = 0.23)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
```

```
-->
  SPEW:   0.04:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
********************************************************************************
```

At this point, the activation of a preference exceeds threshold, and the associated move is selected. This is the indirect move to move the small disk from the left peg to the middle peg.

```
********************************************************************************
                                   CYCLE: 12
================================================================================
LH-EXECUTIVE: 10
--------------------------------------------------------------------------------
SELECT-AMONG-PREFERENCES
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
-->
  SPEW:   0.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

On cycle 13, the selected move is performed and the representations associated with the just-completed iteration of problem solving are suppressed, freeing their resources for other purposes.

```
Move DISK1 from PEG1 to PEG2. (13)

********************************************************************************
                                   CYCLE: 13
```

The partial products of selection are suppressed.

```
================================================================================
LH-EXECUTIVE: 11
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  0.11:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 31 :OPERATOR #<DIRECT-MOVE 21>)
    With: -1.00
```

```
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  0.11:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 30 :OPERATOR #<DIRECT-MOVE 22>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  0.23:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
-->
  SPEW:   0.23:   (PREFERENCE :ID 29 :OPERATOR #<DIRECT-MOVE 23>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  0.68:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
-->
  SPEW:   0.68:   (PREFERENCE :ID 27 :OPERATOR #<DIRECT-MOVE 24>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
-->
  SPEW:   1.00:   (PREFERENCE :ID 25 :OPERATOR #<INDIRECT-MOVE 20>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

The (unselected) direct moves are suppressed.

```
================================================================================
RH-SPATIAL: 4
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 24 :STATE #<PUZZLE 2> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
```

```
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 23 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 22 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 21 :STATE #<PUZZLE 2> :DISK 1 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because all of the preconditions are satisfied, the selected move can be performed. The small disk (1) is transferred from the left peg to the middle peg, producing a new current configuration. This activity takes place in LH-Spatial, which is specialized for the maintenance and transformation of visuospatial representations.

```
================================================================================
LH-SPATIAL: 1
--------------------------------------------------------------------------------
PERFORM-MOVE
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
-->
  SPEW:   0.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

Cycle 14 brings the next iteration of problem solving. Notice that all four centers are active.

```
*******************************************************************************
                                   CYCLE: 14
```

Because the move just performed achieved the goal, the goal can be suppressed.

```
===============================================================================
RH-EXECUTIVE: 3
-------------------------------------------------------------------------------
SUPPRESS-SATISFIED-GOAL
  1.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0 :DISK 2)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
-->
  SPEW:   1.00:   (UNBLOCK-DISK-GOAL :ID 19 :OPERATOR #<DIRECT-MOVE 12> :SOURCE-PEG PEG1 :DEST-
PEG PEG3 :DEST-POS 0 :DISK 2)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because the selected move was successfully applied, the preferred operator marker can be suppressed.

```
===============================================================================
LH-EXECUTIVE: 12
-------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR-MARKER
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
-->
  SPEW:   1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because the selected move was successfully applied, it can be suppressed. At the same time, productions in RH-Spatial propose moves that increase the similarity between the (new) current configuration and the ending configuration.

```
===============================================================================
RH-SPATIAL: 5
-------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 32 :OPERATOR #<INDIRECT-MOVE 20>)
  1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1 :SOURCE-
PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
-->
  SPEW:   1.00:   (INDIRECT-MOVE :ID 20 :STATE #<PUZZLE 2> :GOAL #<UNBLOCK-DISK-GOAL 19> :DISK 1
:SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-POS 0)
    With: -1.00
    Req:  0.00
```

```
    Cont: n/a
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
  1.00:   (PEG :ID 8 :PEG PEG3)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
  1.00:   (PEG :ID 7 :PEG PEG2)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 36 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 8 :PEG PEG3)
-->
```

```
  SPEW:   0.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 6 :PEG PEG1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
```

Because the value of the `*SUPPRESS-OLD-STATES*` variable is `ALL`, the previous state, which has been superseded by the (new) current state, can be suppressed.

```
================================================================================
LH-SPATIAL: 2
--------------------------------------------------------------------------------
SUPPRESS-OLD-STATE
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
-->
  SPEW:   1.00:   (PUZZLE :ID 2 :CONTENTS (2 1) NIL NIL)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
********************************************************************************
```

The heuristic productions in LH-Executive fire, activating preferences for the six proposed moves. Because no goals are currently active, the only heuristics that apply favor legal operators, hill-climbing operators, and the steepest hill-climbing operator.

```
********************************************************************************
                                   CYCLE: 15
================================================================================
LH-EXECUTIVE: 13
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 1)
-->
  SPEW:   0.00:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 0)
```

```
-->
  SPEW:   0.00:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 0.01
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
--------------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 0.03
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
--------------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
    With: 0.03
    Req:  0.03
    Cont: 0.03 (* 1.00 = 0.03)
--------------------------------------------------------------------------------
PREFER-STEEPEST-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 0.05
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
********************************************************************************
```

Over the next few cycles, preferences accrue activation.

```
********************************************************************************
                                   CYCLE: 16
================================================================================
LH-EXECUTIVE: 14
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
```

```
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.09:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
-->
  SPEW:   0.09:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 0.18
    Req:  0.26
    Cont: 0.26 (* 1.00 = 0.26)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.03:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
-->
  SPEW:   0.03:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
    With: 0.05
    Req:  0.08
    Cont: 0.08 (* 1.00 = 0.08)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 1)
  0.01:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
********************************************************************************

********************************************************************************
                                   CYCLE: 17
================================================================================
LH-EXECUTIVE: 15
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.26:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
-->
  SPEW:   0.26:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 0.53
    Req:  0.79
    Cont: 0.79 (* 1.00 = 0.79)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
```

```
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.08:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
-->
  SPEW:   0.08:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
    With: 0.15
    Req:  0.23
    Cont: 0.23 (* 1.00 = 0.23)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 1)
  0.04:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
********************************************************************************

********************************************************************************
                                   CYCLE: 18
================================================================================
LH-EXECUTIVE: 16
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  0.79:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
-->
  SPEW:   0.79:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: 1.58
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.23:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
-->
  SPEW:   0.23:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
    With: 0.45
    Req:  0.68
    Cont: 0.68 (* 1.00 = 0.68)
--------------------------------------------------------------------------------
```

```
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 0)
  0.11:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 1)
  0.11:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
********************************************************************************
```

By cycle 19, a preference accrues an activation that exceeds threshold, and the associated move is selected. This is the direct, ending move that was proposed back on cycle 1 and which originally caused an impasse because one of its preconditions was unsatisfied – transferring the large disk (2) from the left peg to the bottom of the right peg.

```
********************************************************************************
                                   CYCLE: 19
================================================================================
LH-EXECUTIVE: 17
--------------------------------------------------------------------------------
SELECT-AMONG-PREFERENCES
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
-->
  SPEW:   0.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

On cycle 20, the selected move is performed and the representations associated with the just-completed iteration of problem solving are suppressed, freeing their resources for other purposes.

```
Move DISK2 from PEG1 to PEG3. (20)

********************************************************************************
                                   CYCLE: 20
```

The partial products of selection are suppressed.

```
================================================================================
```

```
LH-EXECUTIVE: 18
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  0.34:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 45 :OPERATOR #<DIRECT-MOVE 34>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  0.34:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 44 :OPERATOR #<DIRECT-MOVE 35>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  0.68:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
-->
  SPEW:   0.68:   (PREFERENCE :ID 41 :OPERATOR #<DIRECT-MOVE 38>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
-->
  SPEW:   1.00:   (PREFERENCE :ID 40 :OPERATOR #<DIRECT-MOVE 37>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

The (unselected) direct moves are suppressed.

```
================================================================================
RH-SPATIAL: 6
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 34 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
```

```
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 35 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 36 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG2 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 36 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG2
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 38 :STATE #<PUZZLE 33> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because all of the preconditions are satisfied, the selected move can be performed. The large disk (2) is transferred from the left peg to the right peg, producing a new current configuration.

```
================================================================================
LH-SPATIAL: 3
--------------------------------------------------------------------------------
PERFORM-MOVE
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
-->
  SPEW:   0.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

Cycle 21 brings the final iteration of problem solving.

```
********************************************************************************
```

```
CYCLE: 21
```

Because the selected move was successfully performed, the preferred operator marker can be suppressed.

```
================================================================================
LH-EXECUTIVE: 19
--------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR-MARKER
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
-->
  SPEW:   1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because the selected move was successfully performed, it can be suppressed. At the same time, productions in RH-Spatial propose moves that increase the similarity between the (new) current configuration and the ending configuration.

```
================================================================================
RH-SPATIAL: 7
--------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 46 :OPERATOR #<DIRECT-MOVE 37>)
  1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3 :DEST-
POS 0)
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 37 :STATE #<PUZZLE 33> :DISK 2 :SOURCE-PEG PEG1 :DEST-PEG PEG3
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
PROPOSE-ENDING-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
  1.00:   (PEG :ID 7 :PEG PEG2)
```

```
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 51 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG2
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 5 :DISK 2)
  1.00:   (PEG :ID 6 :PEG PEG1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 8 :PEG PEG3)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 1)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
PROPOSE-TOP-MOVE
  1.00:   (SOLVE-PUZZLE-GOAL :ID 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00:   (DISK :ID 4 :DISK 1)
  1.00:   (PEG :ID 6 :PEG PEG1)
-->
  SPEW:   0.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1
:DEST-POS 0)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
```

Because the value of the `*SUPPRESS-OLD-STATES*` variable is `ALL`, the previous state, which has been superseded by the (new) current state, can be suppressed.

```
================================================================================
LH-SPATIAL: 4
--------------------------------------------------------------------------------
SUPPRESS-OLD-STATE
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
-->
  SPEW:   1.00:   (PUZZLE :ID 33 :CONTENTS (2) (1) NIL)
    With: -1.00
    Req:  0.00
```

```
    Cont: 0.00 (* 1.00 = 0.00)
********************************************************************************
```

The heuristic productions in LH-Executive fire, activating preferences for the five proposed moves. Because all proposed moves are direct, the only heuristics that apply favor legal operators, hill-climbing operators, and the steepest hill-climbing operator.

```
********************************************************************************
                                   CYCLE: 22
================================================================================
LH-EXECUTIVE: 20
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
-->
  SPEW:   0.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: 0.01
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
--------------------------------------------------------------------------------
PREFER-LEGAL-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1 :DEST-
POS 0)
-->
  SPEW:   0.00:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
    With: 0.01
    Req:  0.01
    Cont: 0.01 (* 1.00 = 0.01)
--------------------------------------------------------------------------------
PREFER-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   0.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: 0.03
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
--------------------------------------------------------------------------------
PREFER-STEEPEST-HILL-CLIMBING-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
-->
```

```
  SPEW:   0.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: 0.05
    Req:  0.09
    Cont: 0.09 (* 1.00 = 0.09)
********************************************************************************
```

Over the next few cycles, preferences accrue activation.

```
********************************************************************************
                                   CYCLE: 23
================================================================================
LH-EXECUTIVE: 21
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.09:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
-->
  SPEW:   0.09:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: 0.18
    Req:  0.26
    Cont: 0.26 (* 1.00 = 0.26)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 0)
  0.01:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
-->
  SPEW:   0.01:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
    With: 0.03
    Req:  0.04
    Cont: 0.04 (* 1.00 = 0.04)
********************************************************************************

********************************************************************************
                                   CYCLE: 24
================================================================================
LH-EXECUTIVE: 22
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.26:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
-->
  SPEW:   0.26:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
```

```
    With: 0.53
    Req:  0.79
    Cont: 0.79 (* 1.00 = 0.79)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 0)
  0.04:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
-->
  SPEW:   0.04:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
    With: 0.08
    Req:  0.11
    Cont: 0.11 (* 1.00 = 0.11)
********************************************************************************

********************************************************************************
                                   CYCLE: 25
================================================================================
LH-EXECUTIVE: 23
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  0.79:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
-->
  SPEW:   0.79:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: 1.58
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1 :DEST-
POS 0)
  0.11:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
-->
  SPEW:   0.11:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
--------------------------------------------------------------------------------
ITERATIVELY-ACTIVATE-MA
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 0)
  0.11:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
-->
```

```
  SPEW:   0.11:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
    With: 0.23
    Req:  0.34
    Cont: 0.34 (* 1.00 = 0.34)
********************************************************************************
```

By cycle 26, a preference accrues an activation that exceeds threshold, and the associated move is selected. This is the direct, ending move to transfer the small disk (1) from the middle peg, where it was previously buffered, to the right peg where it resides in the ending configuration.

```
********************************************************************************
                                   CYCLE: 26
================================================================================
LH-EXECUTIVE: 24
--------------------------------------------------------------------------------
SELECT-AMONG-PREFERENCES
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
-->
  SPEW:   0.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

On cycle 27, the selected move is performed and the representations associated with the just-completed iteration of problem solving are suppressed, freeing their resources for other purposes.

```
Move DISK1 from PEG2 to PEG3. (27)

********************************************************************************
                                   CYCLE: 27
```

The partial products of selection are suppressed.

```
================================================================================
LH-EXECUTIVE: 25
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  0.34:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 57 :OPERATOR #<DIRECT-MOVE 48>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  0.34:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
-->
  SPEW:   0.34:   (PREFERENCE :ID 55 :OPERATOR #<DIRECT-MOVE 50>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

```
--------------------------------------------------------------------------------
SUPPRESS-PREFERENCE
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
-->
  SPEW:   1.00:   (PREFERENCE :ID 53 :OPERATOR #<DIRECT-MOVE 49>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

The (unselected) direct moves are suppressed.

```
================================================================================
RH-SPATIAL: 8
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 48 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG1
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (DIRECT-MOVE :ID 51 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG2 :DEST-
POS 1)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 51 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG2
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
--------------------------------------------------------------------------------
SUPPRESS-UNSELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1 :DEST-
POS 0)
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 50 :STATE #<PUZZLE 47> :DISK 2 :SOURCE-PEG PEG3 :DEST-PEG PEG1
:DEST-POS 0)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because all of the preconditions are satisfied, the selected move can be performed. The small disk is transferred from the middle peg to the right peg.

```
================================================================================
```

```
LH-SPATIAL: 5
--------------------------------------------------------------------------------
PERFORM-MOVE
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
-->
  SPEW:   0.00:   (PUZZLE :ID 59 :CONTENTS NIL NIL (2 1))
    With: 1.00
    Req:  1.00
    Cont: 1.00 (* 1.00 = 1.00)
********************************************************************************
```

Cycle 28 marks the end of problem solving.

```
********************************************************************************
                                   CYCLE: 28
```

Because the move was successfully applied, preferences and the preferred operator marker can be suppressed.

```
================================================================================
LH-EXECUTIVE: 26
--------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR-MARKER
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (PUZZLE :ID 59 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because the selected move was successfully performed, it can be suppressed. For the first time, the current and ending configurations are identical, i.e., maximally similar. Therefore, RH-Spatial proposes no new moves.

```
================================================================================
RH-SPATIAL: 9
--------------------------------------------------------------------------------
SUPPRESS-APPLIED-SELECTED-OPERATOR
  1.00:   (SELECTED-OPERATOR :ID 58 :OPERATOR #<DIRECT-MOVE 49>)
  1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3 :DEST-
POS 1)
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
  1.00:   (PUZZLE :ID 59 :CONTENTS NIL NIL (2 1))
-->
  SPEW:   1.00:   (DIRECT-MOVE :ID 49 :STATE #<PUZZLE 47> :DISK 1 :SOURCE-PEG PEG2 :DEST-PEG PEG3
:DEST-POS 1)
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
```

Because the value of the *SUPPRESS-OLD-STATES* variable is ALL, the previous state, which has been superseded by the (new) current state, can be suppressed.

```
================================================================================
LH-SPATIAL: 6
--------------------------------------------------------------------------------
SUPPRESS-OLD-STATE
  1.00:   (PUZZLE :ID 59 :CONTENTS NIL NIL (2 1))
  1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
-->
  SPEW:   1.00:   (PUZZLE :ID 47 :CONTENTS NIL (1) (2))
    With: -1.00
    Req:  0.00
    Cont: 0.00 (* 1.00 = 0.00)
********************************************************************************
```

Because the current configuration has been transformed into the ending configuration, the problem has been solved.

Finally, the summ top-level command is used to print a summary of problem solving. First, for each move, the cycles required to make it and the cycles required to that point in problem solving are displayed. Next, the average capacity utilization of each model center is listed. Finally, the terminal contents of short-term declarative memory are shown. Note that the final puzzle configuration is identical to the ending configuration, demonstrating that the problem has been indeed been solved.

```
? (summ)

NUM     DELTA   TOTAL
1       13      13
2       7       20
3       7       27

LH-EXEC 0.107
RH-EXEC 0.146
LH-SPAT 0.211
RH-SPAT 0.357

================================================================================
all centers filtered by BASE-STATE OPERATOR PREFERENCE SELECTED-OPERATOR BASE-GOAL
--------------------------------------------------------------------------------
  1.00: (END-PUZZLE :ID 3 :CONTENTS NIL NIL (2 1))
  1.00: (SOLVE-PUZZLE-GOAL :ID 1)
  1.00: (PUZZLE :ID 59 :CONTENTS NIL NIL (2 1))
================================================================================
```

APPENDIX C: PARAMETER SETTINGS FOR ALL SIMULATIONS

Each simulation or set of simulations varied on nine dimensions.

(1) 4CAPS model: This indicates whether the normal 4CAPS model, described in Chapter VII, or the chunking model, described in Chapter X, was used. The possible value are *normal* and *chunking*. It is specified in the model code by the `*chunking*` variable.

(2) *preference-scheme* value: This indicates the value of the *preference-scheme* design decision, described in Chapter VII. The possible values are *absolute* and *relative*. It is specified in the model code by the `*preference-scheme*` variable.

(3) *accrual-scheme* value: This indicates the value of the *accrual-scheme* design decision, described in Chapter VII. The possible values are *multiplicative* and *additive*. It is specified in the model code by the `*accrual-scheme*` variable.

(4) *top-moves* value: This indicates the value of the *top-moves* design decision, described in Chapter VII. The possible values are *yes* and *no*. It is specified in the model code by the `*top-moves*` variable.

(5) *top-goal-moves-only* value: This indicates the value of the *top-goal-moves-only* design decision, described in Chapter VII. The possible values are *yes* and *no*. It is specified in the model code by the `*top-goal-moves-only*` variable.

(6) *suppress-old-states* value: This indicates the value of the *suppress-old-states* design decision, described in Chapter VII. The possible values are *all*, *non-goal*, and *none*. It is specified in the model code by the `*suppress-old-states*` variable.

(7) heuristic weights: This indicates the weights of the heuristic productions for selecting between preferences, as described in Chapter VII. The possible values are *optimal* and *errorful*. They correspond to the following sets of weight values, described in Chapter VII.

Heuristic Weight	Optimal Value	Errorful Distribution
w_{legal}	0.0125	U(0, 0.10)
$w_{hill\text{-}climbing}$	0.025	U(0, 0.15)
$w_{steepest\text{-}hill\text{-}climbing}$	0.05	U(0, 0.05)
w_{goal}	0.05	U(0, 0.25)
$w_{top\text{-}goal}$	0.10	U(0, 0.10)

The *optimal* weight values are specified in the model code by the `*w1*`, `*w2*`, `*w3*`, `*w4*`, and `*w5*` variables. The upper bounds of the *errorful*weight distributions are specified by the same variables. In this case, the `*random-weights*` variable indicates that these weights should be interpreted not as constants, but as random variables.

(8) presentation mode: This indicates the mode in which problems are solved. The possible values are *constrained* and *unconstrained*. It is specified in the model code by the

`*constrained*` variable. When the presentation mode is *constrained*, the optimal solution sequence to which the model is constrained is given by the `*constrained-moves*` variable.

(9) center resources: This indicates the resource capacities of the four model centers. The *default* values are 100.0 units for each center. This is sufficiently large that the resource demands of no center exceed its supply for the problems considered in this dissertation project.

(10) notes: Additional comments on the running of simulations.

C1. Individual move time data of Ruiz (1987) and Anderson et al. (1993), Chapter VIII

4CAPS model:	*normal*
preference-scheme value:	{*absolute*, *relative*}
accrual-scheme value:	{*multiplicative*, *relative*}
top-moves value:	{*yes*, *no*}
top-goal-moves-only value:	{*yes*, *no*}
suppress-old-states value:	{*all*, *non-goal*, *none*}
heuristic weights:	*optimal*
presentation mode:	n/a
center resources:	*default*
notes:	The model was run for all 48 combinations of the *preference-scheme*, *accrual-scheme*, *top-moves*, *top-goal-moves-only*, and *suppress-old-states* decisions.

C2. Number of moves data of Anderson et al. (1993), Chapter VIII

4CAPS model:	*normal*
preference-scheme value:	*absolute*
accrual-scheme value:	*multiplicative*
top-moves value:	*yes*
top-goal-moves-only value:	{*yes*, *no*}
suppress-old-states value:	{*all*, *non-goal*, *none*}
heuristic weights:	*errorful*
presentation mode:	*unconstrained*
center resources:	*default*
notes:	The model was run for all 6 combinations of the *top-goal-moves-only* and *suppress-old-states* decisions.

C3. Proportion of problems solved data of Goel et al. (2001), Chapter IX

4CAPS model:	*normal*
preference-scheme value:	*absolute*
accrual-scheme value:	*multiplicative*
top-moves value:	*yes*
top-goal-moves-only value:	{*yes*, *no*}
suppress-old-states value:	*all*

heuristic weights: *errorful*
presentation mode: *unconstrained*
center resources: The resources assigned to each model varied according to two factors, the value of the *top-goal-moves-only* decision and which lesions, if any, were being simulated.

		Frontal Lesions		
Center	Normals	Left	Right	Bilateral
top-goal-moves-only=yes				
LH-Executive	9.0	6.0	9.0	6.0
RH-Executive	5.0	5.0	2.5	2.5
LH-Spatial	100.0	100.0	100.0	100.0
RH-Spatial	100.0	100.0	100.0	100.0
top-goal-moves-only=no				
LH-Executive	9.0	6.0	9.0	6.0
RH-Executive	4.5	4.5	3.5	3.5
LH-Spatial	100.0	100.0	100.0	100.0
RH-Spatial	100.0	100.0	100.0	100.0

notes: The model was run for both values of the *top-goal-moves-only* decision. The temporal mapping between real time and model time was estimated by the procedure described in Chapter IX. It was 2.42 cycles per sec, i.e., the model's allotted time of 290 cycles corresponded to the participants' allotted time of 120 sec.

C4. Number of moves data of Goel et al. (2001), Chapter IX

4CAPS model: *normal*
preference-scheme value: *absolute*
accrual-scheme value: *multiplicative*
top-moves value: *yes*
top-goal-moves-only value: {*yes*, *no*}
suppress-old-states value: *all*
heuristic weights: *errorful*
presentation mode: *unconstrained*
center resources: The resources assigned to each model varied according to two factors, the value of the *top-goal-moves-only* decision and which lesions, if any, were being simulated. They are listed in C3.
notes: The model was run for both values of the *top-goal-moves-only* decision.

C5. Overall solution time data of Goel et al. (2001), Chapter IX

4CAPS model: *normal*
preference-scheme value: *absolute*
accrual-scheme value: *multiplicative*
top-moves value: *yes*
top-goal-moves-only value: {*yes*, *no*}

suppress-old-states value:	*all*
heuristic weights:	*errorful*
presentation mode:	*unconstrained*
center resources:	The resources assigned to each model varied according to two factors, the value of the *top-goal-moves-only* decision and which lesions, if any, were being simulated. They are listed in C3.
notes:	The model was run for both values of the *top-goal-moves-only* decision.

C6. Individual move time data of Anderson et al. (2005), Chapter X

4CAPS model:	{*normal*, *chunking*}
preference-scheme value:	*absolute*
accrual-scheme value:	*multiplicative*
top-moves value:	*yes*
top-goal-moves-only value:	*no*
suppress-old-states value:	*all*
heuristic weights:	*optimal*
presentation mode:	n/a
center resources:	*default*
notes:	none

C7. Neuroimaging data of Anderson et al. (2005), Chapter X

4CAPS model:	*normal*
preference-scheme value:	*absolute*
accrual-scheme value:	*multiplicative*
top-moves value:	*yes*
top-goal-moves-only value:	{*yes*, *no*}
suppress-old-states value:	{*all*, *non-goal*, *none*}
heuristic weights:	*optimal*
presentation mode:	n/a
center resources:	LH-Executive and RH-Executive were each assigned 13.0 units of resources. LH-Spatial and RH-Spatial were each assigned 10.0 units.
notes:	The model was run for all 6 combinations of the *top-goal-moves-only* and *suppress-old-states* decisions.

ENDNOTES

1. There is a mistake in the list of problems presented in the published report. Specifically, the 5-disk problem [(5 4 3 2 1) () ()] => [(5 4 1) (3) (2)] requires only 5 moves to solve. This appears to be a simple misprint because two permutations of the pegs of the ending configuration yield problems that requires the full 31 moves to solve. The following one was chosen arbitrarily: [(5 4 3 2 1) () ()] => [(2) (5 4 1) (3)]

2. Actually, the temporal measure computed was the average time per move, which is the overall solution time divided by the number of moves. The measure used here, overall solution time, was computed as the product of the average time per move and the number of moves.

3. Also tested were patients with left temporal lesions and patients with right temporal lesions. These two groups are ignored here because the model does not contain centers corresponding to these brain areas, and because no interesting results were found for them.

4. These data are presented figurally in the published paper. Unfortunately, their numerical values are no longer available (Morris, personal communication).

5. Once again, patients with left temporal lesions and patients with right temporal lesions were also tested. These two groups are ignored here because the model does not contain centers corresponding to these brain areas, and because no interesting results were found for them.

6. Once again, these data are presented figurally in the published paper. Unfortunately, their numerical values are no longer available (Morris, personal communication).

7. This effect disappeared when the times were log-transformed.

8. These data are plotted figurally in the published report. I requested the numerical values from the corresponding author. Unfortunately, they were not provided during the time when I worked on this dissertation project.

9. I requested the elided error data from the corresponding author. Unfortunately, they were not provided during the time when I worked on this dissertation project.

10. These data are plotted figurally in the published report. I requested the numerical values from the corresponding author. Unfortunately, they were not provided until after this dissertation project was completed.

11. Goel et al. (2001) reconstrued working memory limitations as the product of time-based decay, not a fixed resource supply. This difference is not important for the present discussion.

12. There is currently a large effort to identify the neural bases of the BOLD response (e.g., Logothetis, Pauls, Augath, Trinath, & Oeltermann, 2001). Capacity utilization is an intentionally molar concept, a working hypothesis that enables progress today while the biological details are still being worked out.

13. I requested an example problem and numerical values of the data from the corresponding author. Unfortunately, they were not provided during the time when I worked on this dissertation project.

14. It should be noted that the data did not offer strong guidance on the value of this decision. However, there were a few occasions when a *no* value of the *top-goal-moves-only* decision yielded higher (if not *reliably* higher) correlations to the data than a *yes* value: the number of moves data that Anderson et al. (1993) collected from normal young adults (i.e., Table 18), the proportion of problems solved data that Goel et al. (2001) collected from intact normals and frontal patients (i.e., Table 21), and the left frontal activation data that Anderson et al. (2005) collected from normal young adults (i.e., Table 24).

15. It should be noted that a recent ACT-R model of TOH problem solving (Gunzelmann & Anderson, 2003) does address some of the data on practice effects. This model was not considered because it does not make contact with any of the data sets that are at the heart of this dissertation project (Anderson et al., 1993; Anderson et al., 2005; Fincham et al., 2002; Goel et al., 2001; Ruiz, 1987).

REFERENCES

Aguirre, G. K., Zarahn, E., & D'Esposito, M. (1998). The variability of human, BOLD hemodynamic responses. *NeuroImage, 8*, 360-369.

Ahlum-Heath. M. E., & Di Vesta, F. J. (1986). The effect of conscious controlled verbalization of a cognitive strategy on transfer in problem solving. *Memory & Cognition*, *14*, 281-285.

Alivisatos, B., & Petrides, M. (1997). Functional activation of the human brain during mental rotation. *Neuropsychologia*, *35*, 111-118.

Altmann, E. M., & Trafton, J. G. (2002). Memory for goals: An activation-based mode. *Cognitive Science*, *26*, 39-83.

Aman, C. J., Roberts, R. J., & Pennington, B. F. (1998). A neuropsychological examination of the underlying deficit in attention deficit hyperactivity Disorder: Frontal lobe versus right parietal lobe theories. *Developmental Psychology*, *34*, 956-969.

Anderson, J. R. (1976). *Language, memory, and thought*. New York: Wiley.

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

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.

Anderson, J. R., Albert, M. V., & Fincham, J. M. (2005). Tracing problem solving in real time: fMRI analysis of the subject-paced Tower of Hanoi. *Journal of Cognitive Neuroscience*, *17*, 1261-1274.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*, 1036-1060.

Anderson, J. R., & Douglass, S. (2001). Tower of Hanoi: Evidence for the cost of goal retrieval. *Journal of Experimental Psychology: Learning, Memory, & Cognition*, *27*, 1331-1346.

Anderson, J. R., Kushmerick, N., & Lebiere, C. (1993). The Tower of Hanoi and goal structures. In J. R. Anderson (Ed.), *Rules of the Mind* (pp. 121- 142). 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.

Arbib, M. A., Billard, A., Iacoboni, M., & Oztop, E. (2000). Synthetic brain imaging: grasping, mirror neurons and imitation. *Neural Networks*, *13*, 975-997.

Baddeley, A. D. (1996). Exploring the central executive. *Quarterly Journal of Experimental Psychology*, *49A*, 5-28.

Baddeley, A. (2003). Working memory: Looking bask and looking forward. *Nature Reviews Neuroscience*, *4*, 829-839.

Badre, D., & Wagner, A. D. (2002). Semantic retrieval, mnemonic control, and prefrontal cortex. *Behavioral and Cognitive Neuroscience Reviews*, *1*, 206-218.

Barsalou, L. W. (1983). Ad hoc categories. *Memory & Cognition*, *11*, 211-227.

Bishop, D. V. M., Aamodt-Leeper, G., Creswell, C., McGurk, R., & Skuse, D. H. (2001). Individual differences in cognitive planning on the Tower of Hanoi task: Neuropsychological maturity or measurement error? *Journal of Child Psychology and Psychiatry*, *42*, 551-556.

Borys, S. V., Spitz, H. H., & Dorans, B. A. (1982). Tower of Hanoi performance of retarded young adults and nonretarded children as a function of solution length and goal state. *Journal of Experimental Child Psychology*, *33*, 87-110.

Boynton, G. M., Engel, S. A., Glover, G. H., & Heeger, D. J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1. *The Journal of Neuroscience, 16*, 4207-4221.

Brennan, M., Welsh, M. C., & Fisher, C. B. (1997). Aging and executive function skills: An examination of a community-dwelling older population. *Perceptual and Motor Skills*, *84* 1187-1197.

Buckner, R. L., Corbetta, M., Schatz, J., Raichle, M. E., & Petersen, S. E. (1996). Preserved speech abilities and compensation following prefrontal damage. *Proceedings of the National Academy of Science USA*, *93*, 1249-1253.

Bunge, S. A., Wendelken, C., Badre, D., & Wagner, A. D. (2005). Analogical reasoning and prefrontal cortex: Evidence for separable retrieval and integration mechanisms. *Cerebral Cortex*, *15*, 239-249.

Bustini, M., Stratta, P., Daneluzzo, E., Pollice, R., Prosperini, P., & Rossi, A. (1999). Tower of Hanoi and WCST performance in schizophrenia: problem-solving capacity and clinical correlates. *Journal of Psychiatric Research*, *33*, 285-290.

Byrne, M. D., & Bovair, S. (1997). A working memory model of a common procedural error. *Cognitive Science*, *21*, 31-61.

Calvert, G. A., Brammer, M. J., Morris, R. G., Williams, S. C. R., King, N., & Matthews, P. M. (2000). Using fMRI to study recovery from acquired dysphasia. *Brain and Language*, *71*, 391-399.

Cao, Y., Vikingstad, E. M., George, K. P., Johnson, A. F., & Welch, K. M. A. (1999). Cortical language activation in stroke patients recovering from aphasia with functional MRI. *Stroke*, *30*, 2331-2340.

Carpenter, P. A., Just, M. A., & Shell, P. (1990). What one intelligence test measures: A theoretical account of the processing in the Raven Progressive Matrices Test. *Psychological Review*, *97* 404-431.

Cooper, R., & Shallice, T. (1995). Soar and the case for unified theories of cognition. *Cognition*, *55*, 115-149.

Cooper, R., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, *17*, 297-338.

Corbetta, M., & Shulman, G. L. (2002). Control of goal-directed and stimulus-directed attention in the brain. *Nature Reviews Neuroscience*, *3*, 201-215.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2001). *Introduction to algorithms* (3^{rd} ed.). Cambridge, MA: MIT Press.

Curtis, C. E., & D'Esposito, M. (2003). Persistent activity in the prefrontal cortex during working memory. *TRENDS in Cognitive Sciences*, *7*, 414-423.

Dantzig, G. B., & Thapa, M. N. (1997). *Linear programming 1: Introduction*. New York: Springer-Verlag.

Davies, S. P. (2000). Move evaluation as a predictor and moderator of success in solutions to well-structured problems. *Quarterly Journal of Experimental Psychology*, *53A*, 1186-1201.

Dehaene, S. & Changeux, J.-P. (1997). A hierarchical neuronal network for planning behavior. *Proceedings of the National Academy of Science USA*, *94*, 13293-13298.

Duncan, J., Seitz, R. J., Kolodny, J., Bor, D., Herzog, H., Ahmed, A., Newell, F. N., & Emslie, H. (2000). A neural basis for general intelligence. *Science, 289*, 457-460.

Elfgren, C. I., & Risberg, J. (1998). Lateralized frontal blood flow increases during fluency tasks: Influence of cognitive strategy. *Neuropsychologia*, *36*, 505-512.

Erman, L. D., Hayes-Roth, F., Lesser, V. R., & Reddy, D. R. (1980). The Hearsay-II speech understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, *12*, 213-253.

Ewert, P. H., & Lambert, J. F. (1932). The effect of verbal instructions upon the formation of a concept. *Journal of General Psychology*, *6*, 400-411.

Fodor, J. A. (1983). *The modularity of mind*. Cambridge, MA: MIT Press.

Fincham, J, M., Carter, C. S., van Veen, V., Stenger, V. A., & Anderson, J. R. (2002). Neural mechanisms of planning: A computational analysis using event-related fMRI. *Proceedings of the National Academy of Sciences*, *99*, 3346-3351.

Frith, C. D., Friston, K., Liddle, P. F., & Frackowiak, R. S. J. (1991). Willed action and the prefrontal cortex in man: A study with PET. *Proceedings of the Royal Society of London B*, *244*, 241-246.

Gagné, R. M., & Smith, E. C. (1962). A study of the effects of verbalization on problem solving. *Journal of Experimental Psychology*, *63*, 12-18.

Gleissner, U., & Elger, C. E. (2001). The hippocampal contribution to verbal fluency in patients with temporal lobe epilepsy. *Cortex*, *37*, 55-63.

Glosser, G., & Goodglass, H. (1990). Disorders in executive control functions among aphasic and other brain-damaged patients. *Journal of Clinical and Experimental Neuropsychology*, *12*, 485-501.

Goel, V. (2003). Evidence for dual neural pathways for syllogistic reasoning. *Psychologica*, *32*, 301-309.

Goel, V. & Grafman, J. 1995. Are the frontal lobes implicated in "planning" functions? Interpreting data from the Tower of Hanoi. *Neuropsychologia*, *33*, 623-642.

Goel, V., Pullara, S. D., & Grafman, J. (2001). A computational model of frontal lobe dysfunction: working memory and the Tower of Hanoi task. *Cognitive Science*, *25*, 287-313.

Goldman, S. R., & Varma, S. (1995). CAPping the construction-integration model of discourse comprehension. In C. Weaver, S. Mannes, & C. Fletcher (Eds.), *Discourse comprehension: Essays in honor of Walter Kintsch* (pp. 337-358). Hillsdale, NJ: Erlbaum.

Gourovitch, M. L., Kirkby, B. S., Goldberg, T. E., Weinberger, D. R., Gold, J. M., Esposito, G., Van Horn, J. D., & Berman, K. F. (2000). *Neuropsychology*, *14*, 353-360.

Gray, J. R., Chabris, C. F., & Braver, T. S. (2003). Neural mechanisms of general fluid intelligence. *Nature Neuroscience*, *6*, 316-322.

Gunzelmann, G., & Anderson, J. R. (2003). Problem solving: Increases planning with practice. *Cognitive Systems Research*, *4*, 57-76.

Haarmann, H. J., Just, M. A., & Carpenter, P. A. (1997). Aphasic sentence comprehension as a resource deficit: A computational approach. *Brain and Language, 59*, 76-120.

Harel, D. (1987). *Algorithmics: The spirit of computing*. Workingham, England: Addison-Wesley.

Hayes, J. R., & Simon, H. A. (1974). Understanding written problem instructions. In L. W. Gregg (Ed.), *Knowledge and cognition* (pp. 167-200). Hillsdale, NJ: Erlbaum.

Hayes, J. R., & Simon, H. A. (1977). Psychological differences among problem isomorphs. In N. J. Castellan, D. B. Pisoni, and G. R. Potts (Eds.), *Cognitive theory, Volume 2* (pp. 21-41). Hillsdale, NJ: Erlbaum.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart, J. L. McClelland, & the PDP Research Group (Eds.), *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations* (pp. 77-109). Cambridge, MA: MIT Press.

Hofstadter, D., & The Fluid Analogies Research Group. (1995). *Fluid concepts and creative analogies: Computer models of the fundamental mechanisms of thought*. New York: Basic Books.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., & Thagard, P. R. (1986). *Induction: Processes of inference, learning, and discovery*. Cambridge, MA: The MIT Press.

Horwitz, B., & Tagamets, M.-A. (1999). Predicting human functional maps with neural net modeling. *Human Brain Mapping*, *8*, 137-142.

Huguenard, B. R., Lerch, F. J., Junker, B. W., Patz, R. J., & Kass, R. E. (1997). Working Memory failure in phone-based interaction. *ACM Transactions on Computer-Human Interaction*, *4*, 67-102.

Just, M. A., & Carpenter, P. A. (1985). Cognitive coordinate systems: Accounts of mental rotation and individual differences in spatial ability. *Psychological Review*, *92*, 137-172.

Just, M. A., & Carpenter, P. A. (1987). *The psychology of reading and language comprehension*. Boston: Allyn and Bacon.

Just, M. A., & Carpenter, P. A. (1992). A capacity theory of comprehension: Individual differences in working memory. *Psychological Review*, *99*, 122-149.

Just, M. A., Carpenter, P. A., & Hemphill, D. D. (1996). Constraints of processing capacity: Architectural or implementational? In D. Steier & T. M. Mitchell (Eds.), *Mind matters: A tribute to Allan Newell* (pp. 141-178). Hillsdale, NJ: Erlbaum.

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping*, *8*, 128-136.

Just, M. A., & Varma, S. (2002). A hybrid architecture for working memory: Reply to MacDonald and Christiansen (2002). *Psychological Review*, *109*, 55-65.

Just, M. A., & Varma, S. (2006). *The organization of thinking: What functional brain imaging reveals about the neuroarchitecture of complex cognition*. Manuscript submitted for publication.

Kahneman, D. (2003). A perspective on judgment and choice. *American Psychologist*, *58*, 697-720.

Karbe, H., Thiel, A., Weber-Luxenburger, G., Herholz, K., Kessler, J., & Heiss, W.-D. (1998). Brain plasticity in poststroke aphasia: What is the contribution of the right hemisphere? *Brain and Language*, *64*, 215-230.

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

Knobloch, C. A. (1991). Search reduction in hierarchical problem solving. In *Proceedings of the Ninth National Conference on Artificial Intelligence*. Menlo Park, CA: AAAI Press.

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

Lehto, J. (1996). Are executive function tests dependent on working memory capacity? *The Quarterly Journal of Experimental Psychology*, *49A*, 29-50.

Logothetis, N.K., Pauls, J., Augath, M. A., Trinath, T., & Oeltermann, A. (2001). Neurophysiological investigation of the basis of the fMRI signal. *Nature*, *412*, 150-157.

Luce, R. D. (1986). *Response times: Their role in inferring mental organization.* Oxford: Oxford University Press.

Luria, A. R. (1966). *Higher cortical functions in man*. London: Tavistock.

Meyer, D. E., & Kieras, D. E. (1997). A computational theory of executive cognitive processes and multiple-task performance: I. Basic mechanisms. *Psychological Review*, *104*, 3-65.

Milner, B. (1982). Some cognitive effects of frontal-lobe lesions in man. *Philosophical Transactions of the Royal Society of London B*, *298*, 211-226.

Minsky, M. (1985). *Society of mind*. New York: Simon & Schuster.

Morris, R. G., Miotto, E. C., Feigenbaum, J. D., Bullock, P., & Polkey, C. E. (1997a). The effect of goal-subgoal conflict on planning after frontal- and temporal-lobe lesions in humans. *Neuropsychologia*, *35*, 1147-1157.

Morris, R. G., Miotto, E. C., Feigenbaum, J. D., Bullock, P., & Polkey, C. E. (1997b). Planning ability after frontal and temporal lobe lesions in Humans: The effects of selection quivocation and working memory load. *Cognitive Neuropsychology*, *14*, 1007-1027.

Ng, V. W. K., Eslinger, P. J., Williams, S. C. R., Brammer, M. J., Bullmore, E. T., Andrew, C. M., Suckling, J., Morris, R. G., & Benton, A. L. (2000). Hemispheric preference in visuospatial processing: A complementary approach with fMRI and lesion studies. *Human Brain Mapping*, *10*, 80-86.

Newell, A. (1973). You can't play 20 questions with nature and win: Projective comments on the papers in this symposium. In W. G. Chase (Ed.), *Visual information processing* (283-308). New York: Academic Press.

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

Newell, A., & Simon, H. A. (1963). GPS, a program that simulates human thought. In E. Feigenbaum & J. Feldman (Eds.), *Computers and thought* (pp. 279-293). Cambridge, MA: MIT Press.

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

Newman, S. D., Carpenter, P. A., Varma, S., & Just, M. A. (2003). Frontal and parietal participation in problem solving in the Tower of London: fMRI and computational modeling of planning and high-level perception. *Neuropsychologia*, *41*, 1668-1682.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In R. J. Davidson, G. E., Schwartz, & D. E. Shapiro (Eds.), *Consciousness and self-regulation* (Vol. 4) (pp. 1-18). New York: Plenum Press.

Parks, R. W., & Cardoso, J. (1997). Parallel distributed processing and executive functioning: Tower of Hanoi neural networks model in healthy controls and left frontal lobe patients. *International Journal of Neuroscience*, *89*, 217-240.

Phelps, E. A., Hyder, F., Blamire, A. M., & Shulman, R. G. (1997). fMRI of the prefrontal cortex during overt verbal fluency. *NeuroReport*. *8*, 561-565.

Polk, T. A., Simen, P., Lewis, R. L., & Freedman, E. (2002). A computation approach to control in complex cognition. *Cognitive Brain Research*, *15*, 71-83.

Piaget, J. (1976). *The grasp of consciousness*. Cambridge, MA: MIT Press.

Rönnlund, M., Lövdén, M., & Nilsson, L.-G. (2001). Adult age differences in Tower of Hanoi performance: Influence from demographic and cognitive variables. *Aging, Neuropsychology, and Cognition*, *8*, 269-283.

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

Ruiz, D., & Newell, A. (1989/1993). Tower-Noticing Triggers Strategy-Change in the Tower of Hanoi: A Soar Model. In P. S. Rosenbloom, J. E. Laird, & A. Newell (Eds.), *The Soar papers: Research on Integrated Intelligence, Volume Two* (pp. 934-941). Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. & the PDP Research Group. (Eds.). (1986). *Parallel distributed computing: Explorations in the microstructure of cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.

Rumelhart, D. E., & McClelland, J. L. (1982). An interactive activation model of context effects in letter perception: Part 2. The contextual enhancement effect and some tests and extension of the model. *Psychological Review*, *89*, 60-94.

Schank, R. C. (1982). *Dynamic memory: A theory of reminding and learning in computers and people*. New York: Cambridge University Press.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Erlbaum.

Shallice, T. (1982). Specific impairments of planning. *Philosophical Transactions of the Royal Society of London B*, *298*, 199-209.

Shallice, T. (1988). *From neuropsychology to mental structure*. Cambridge: Cambridge University Press.

Shallice, T., & Burgess, P. (1996). The domain of supervisory processes and temporal organization of behaviour. *Philosophical Transactions of the Royal Society of London B*, *351*, 1405-1412.

Shallice, T., & Evans, M. E. (1978). The involvement of the frontal lobes in cognitive estimation. *Cortex*, *14*, 294-303.

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

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

Simon, H. A., & Newell, A. (1961). Computer simulation of human thinking and problem solving. *Datamation*, *7*, 18-20.

Sloman, S. A. (1996). The empirical case for two systems of reasoning. *Psychological Bulletin*, *119*, 3-22.

Spitz, H. H., Minsky, S. K., & Bessellieu, C. L. (1984). Subgoal length versus full solution length in predicting Tower of Hanoi problem-solving performance. *Bulletin of the Psychonomic Society*, *22*, 301-304.

Spitz, H. H., Minsky, S., & Bessellieu, C. L. (1985). Influence of planning time and first-move strategy on Tower of Hanoi problem-solving performance of mentally retarded young adults and nonretarded children. *American Journal of Mental Deficiency*, *90*, 46-56.

Spitz, H. H., Webster, N. A., & Borys, S. V. (1982). Further studies of the Tower of Hanoi problem-solving performance of retarded young adults and nonretarded children. *Developmental Psychology*, *18*, 922-930.

Squire, L. R., & Zola-Morgan, S. (1991). The medial temporal lobe memory system. *Science*, *253*, 1380-1386.

Stanovich, K. E., & West, R. F. (2000). Individual differences in reasoning: Implications for the rationality debate? *Behavioral and Brain Sciences*, *23*, 645-726.

Thibadeau, R., Just, M. A., & Carpenter, P. A. (1982). A model of the time course and content of reading. *Cognitive Science*, *6*, 157-203.

Thulborn, K. R., Carpenter, P. A., & Just, M. A. (1999). Plasticity of language-related brain function during recovery from stroke. *Stroke*, *30*, 749-754.

Trojano, L., Grossi, D., Linden, D. E. J., Formisano, E., Goebel, R., Cirillo, S., Elefante, R., Di Salle, F. (2002). Coordinate and categorical judgments in spatial imagery. An fMRI study. *Neuropsychologia*, *40*, 1666-1674.

Troyer, A. K., Moscovitch, M., & Winocur, G. (1997). Clustering and switching as two components of verbal fluency: Evidence from younger and older healthy adults. *Neuropsychology*, *11*, 138-146.

Troyer, A. K., Moscovitch, M., Winocur, G., Alexander, M. P., & Stuss, D. (1998). Clustering and switching on verbal fluency: The effects of focal frontal- and temporal-lobe lesions. *Neuropsychologia*, *36*, 499-504.

Tversky, A., & Kahneman, D. (1973). Availability: A heuristic for judging frequency and probability. *Cognitive Psychology*, *5*, 207-232.

VanLehn, K. (1991). Rule acquisition events in the discovery of problem-solving strategies. *Cognitive Science*, *15*, 1-47.

Wood, J. N., & Grafman, J. (2003). Human prefrontal cortex: processing and representational perspectives. *Nature Reviews Neuroscience*, *4*, 139-147.

Zhang, J., & Norman, D. A. (1994). Representations in distributed cognitive tasks. *Cognitive Science*, *18*, 87-122.