# The Role of Open Goals in Acquiring Problem Relevant Information

A Dissertation Submitted to the
**Department of Psychology**
**Carnegie Mellon University**
in Partial Fulfillment of the
Requirements for the Degree of
**Doctor of Philosophy**

by

Jarrod Moss

April 21, 2006

## Abstract

There have been a number of recent findings that indicate that open goals influence cognition even when the current task has no relation to the one where the goal was originally set. An open goal is a goal which has been set but one for which the task associated with it has not been completed. The purpose of the work presented in this dissertation was to explore what effects open goals may have on problem solving. In particular, based on findings in problem solving and other areas of cognitive psychology it was hypothesized that open goals would influence what information entered the problem-solving process. A series of studies was designed to establish the effect of open goals on the acquisition and use of problem relevant information from sources outside of the original problem-solving task. It was found that problem relevant information, or hints, presented in a second task in between attempts at solving problems aided problem solving. The participants in the studies were not informed that they would be seeing hints, and most of the participants did not notice any relationship between the problem-solving task and the second task where the hints for the problems were embedded. It was found that presenting hints was more effective when an open goal was present than if the hint was presented before problem solving began.

Another set of studies was conducted to examine properties of this open goal mechanism. One study found that there is no association between working memory limits and the number of open goals which influence information acquisition. There was also no indication of a limit on the number of open goals which can influence information acquisition. Another study collected concurrent verbal protocols to determine when people reached an impasse in problem solving. The presented hints were only effective in aiding problem solving for which no impasse had been reached. The results of these studies help to constrain the development of a mechanistic theory of how open goals influence information acquisition in problem solving. The implications of this work for problem-solving research are discussed including potential contributions to our understanding of insight, incubation, transfer, and creativity.

## Acknowledgments

I would like to thank Ken Kotovsky and Jon Cagan for being wonderful advisors throughout my time at Carnegie Mellon. They allowed me the freedom to pursue my ideas while also offering constant guidance and support. I have learned a great deal from them over the years that will continue to be useful in both my academic and personal lives.

I would like to thank my committee members, David Klahr and Marsha Lovett, for the many discussions of research ideas and for your advice and guidance.

I would like to thank the many students who have helped me with my research over the years. In particular, I would like to thank Krista Pasfield for all of our discussions and her work over the past two years. I would also like to thank Azim Ali and Andrea Price for their contributions to the research presented here.

I have had the privilege of having an outstanding group of friends in graduate school. I would like to thank Julie Booth, Rachel Diana, and Sarah Novak for their friendship and support over the years.

I could not have made it this far without the support of my wife, Jami. She has been a wonderful partner over the years, and I look forward to her continuing love and support for many years to come.

My family has been a constant source of support over the years. I would like to thank them for making all of this possible and for always believing in me.

# Table of Contents

## Table of Figures

# Chapter 1

# Introduction

## 1.1 Overview

Many aspects of human problem solving are now well understood, and much of this understanding is based on understanding how people work within a certain problem space in order to solve a problem (Newell & Simon, 1972). In extending the theory of problem solving to more complex and ill-structured problems Simon (1973) noted that in solving such problems there must be a noticing and evoking mechanism that brought relevant information into the immediate problem-solving process at the right time. This mechanism must be able to notice relevant information from the environment or evoke it from memory. Such a mechanism may also play a role in solving insight problems requiring a representation change as people seem to be guided in their search for a new representation by noticing certain invariants in their repeated failures to solve a problem (Kaplan & Simon, 1990). These examples highlight the role of noticing relevant information during problem solving, and they indicate the importance of understanding under what conditions information is likely to be noticed and incorporated into problem solving. There are a number of variables that may influence which information gets noticed, and one likely influence is the presence of open problem-solving goals.

Zeigarnik (1927/1938) originally demonstrated that interrupted tasks were recalled more readily than completed ones. A number of studies have attempted to replicate this finding with a variety of tasks, but the results of these replications have been mixed (see Van Bergen, 1968, for a review). More recently, there have been a number of studies which demonstrate that unsolved problems or open goals in general are maintained in an accessible state and may exert influences on other tasks (Marsh, Hicks, & Brink, 1998; Patalano & Seifert, 1994; Rothermund, 2003; Seifert & Patalano, 1991; Shah & Kruglanski, 2002). I define an open goal as a goal which has been set but one for which the task associated with it has not been completed. In the work presented here open goals are synonymous with unsolved problems. However, there is evidence from domains outside of problem solving that open goals may influence cognition more generally, and so I have defined open goals to be a more general construct than simply unsolved problems.

The thesis that is explored in this work is that open goals influence the acquisition of problem relevant information. The purpose of the work presented here is to examine the mechanism by which open goals are maintained at a high level of accessibility, and how these goals influence the acquisition of problem relevant information. Understanding this mechanism will help in understanding how people solve complex and ill-structured problems as well as contribute to our understanding of a number of problem-solving phenomena such as insight, incubation, transfer, representation change, and creativity.

The next section reviews some recent findings that indicate that open goals influence cognition across a variety of tasks including memory, task switching, and problem solving. The purpose of the section is to provide a brief overview of some of the different areas where the influence of open goals has been directly studied in order to demonstrate how general this mechanism appears to be. The following sections explore how open goals relate to findings in the problem-solving literature including insight, incubation, transfer, and creativity. Some ideas for how an open goal mechanism may operate in problem solving more broadly are presented followed by a discussion of how open goals may relate to other cognitive processes. This chapter concludes by providing an overview of the four studies presented in the dissertation.

## 1.2 The Influence of Open Goals

One of the earliest examples of an influence of open goals comes from Zeigarnik's work showing that tasks that were interrupted were more likely to be recalled than tasks which participants were allowed to finish (Zeigarnik, 1927/1938). A recent set of replications of the Zeigarnik effect generally found that unsolved problems were recalled better than solved problems, but that the relative number of unsolved to solved problems played a role in whether there was a recall advantage for unsolved problems (Patalano & Seifert, 1994; Seifert, Meyer, Davidson, Patalano, & Yaniv, 1995; Seifert & Patalano, 1991). In particular it was found that set size affected recall of unsolved problems such that an increased number of unsolved problems led to worse recall for unsolved problems, but the set size of solved problems was not associated with percent recall for solved problems. Additionally, there was some evidence that reaching an impasse may be necessary for problem solvers to show a Zeigarnik effect. Overall, these results support the idea that open goals have some kind of special status in memory.

Other findings related to recall in memory show that items in memory for which there has been a failure to recall the item are more likely to be recalled successfully after a subsequent presentation of the item in a second task than are items which are presented without a previous failure (Seifert et al., 1995; Yaniv & Meyer, 1987; Yaniv, Meyer, & Davidson, 1995). These results led the authors to propose a memory sensitization hypothesis where items for which a retrieval failure has occurred are marked in some way that makes it more likely that further exposures to the marked item will lead to a recognition of the relationship to the previous retrieval failure and consequent assimilation of the item so that further attempts at recall will be more likely to succeed.

Open goals also interfere with performance on other tasks. Shah and Kruglanski (2002) found that anagram solving performance was affected by a subliminal prime related to an alternative task goal. Their anagrams each had multiple solutions and participants were asked to find as many solutions as possible for each anagram. They found that when this prime was presented during the anagram task participants spent less time trying to generate solutions for each anagram and generated fewer answers for the anagrams. They also found that the effect of the prime depended on whether the alternative task was perceived as related to the anagram task. Their results demonstrate that unrelated open goals can interfere with other tasks.

Open problem-solving goals can affect not only other problem-solving activities but also affect response time when switching between two tasks unrelated to the original problem-solving activity. Rothermund (2003) had subjects perform difficult multiple choice synonym problems and manipulated whether the participants were told their answers were correct or incorrect. The problems were not presented again, and they were difficult enough that participants appeared to accept the feedback as true. In a second pair of tasks performed after the synonym problems, participants had to read words while also detecting a tone on certain trials. The word to be read was placed between two distractor words, and on some trials the distractor words were words from the synonym problems. Distractors from "incorrect" problems increased response time to the secondary task of the dual tasks relative to unrelated distractors, while distractors from "correct" problems decreased response time to the secondary task. This result held even if the priority of the subsequent dual tasks was switched. The author argues that the most likely time to find interference from failed goals is when goals are being switched such as when participants are transitioning between the two tasks of reading the word and responding to the tone. The finding that open goals interfere with task transitions could indicate that the open goal is competing for attention with other ongoing tasks once relevant information has been identified.

The entire area of research on prospective memory can be viewed as studying the situation where someone has an open goal to perform some action. Research on these yet to be completed intentions shows a persistent level of activation so that items related to these intentions are recognized faster than other unrelated items (Goschke & Kuhl, 1993; Marsh, Hicks, & Bink, 1998). In addition Marsh et al. (1998) also found that items related to completed intentions took longer to recognize than items unrelated to intentions. This set of results are similar to those by Rothermund (2003) described above who used incorrectly answered problems instead of intended actions. The studies described in this section demonstrate that open goals influence processing across a number of different areas including memory, attention, and problem solving. These findings illustrate the generality of the influence of open goals. The following sections explore how open goals relate to areas of problem solving including insight, incubation, transfer, and creativity.

## 1.3 Insight

The sudden flash of insight that people commonly report when solving difficult problems is an interesting aspect of problem solving, but it has proven difficult to explain. There have been a number of theories which have attempted to explain this phenomenon with varying degrees of success (Sternberg & Davidson, 1995). One aspect of this phenomenon is that the information seemingly pops into mind and does not seem to necessarily follow from the train of thought the person was pursuing. It is often difficult to say where insights come from or how people arrive at them since methods like protocol analysis often fail to reveal the source of insights as people tend to stop talking during the critical moments (Schooler, Ohlsson, & Brooks, 1993). There are a number of different views of insight including some which hold that it is special and not explainable in terms of ordinary cognitive processes. The view taken in this dissertation is that the processes can be explained and are related to, if not the same as, processes that operate in all kinds of problem-solving situations. This way of looking at insight as related to or along a continuum with normal problem solving is similar to views that have been expressed in the literature (e.g., Kotovsky, 2003; Weisberg, 1988).

### *1.3.1 Opportunistic Assimilation*

One idea for how information can aid problem solving during which an impasse has been reached is the idea of opportunistic assimilation outlined by Seifert, Meyer, Davidson, Patalano, and Yaniv (1995). Their idea is that when impasses are encountered a set of failure indices are setup in memory which link to the problem. Encounters with information in the environment may lead to retrieval of these indices when the information is related to the problem. This new information may lead to overcoming the impasse and experiencing insight. They presented two sources of data to support their hypothesis. One was their replication of the Zeigarnik effect which they used to argue that reaching an impasse was the key to increased accessibility of unsolved problems relative to solved problems (Patalano & Seifert, 1994; Seifert & Patalano, 1991).

Seifert et al. (1995) also supported their hypothesis with data from a series of studies examining retrieval failures and tip of the tongue experiences (Yaniv & Meyer, 1987; Yaniv et al., 1995). In these studies participants were given the definition of a low frequency word and asked to recall it, and if they could not recall the word they were asked to rate their feeling of knowing (FOK) on a scale of 1-5. The definition task was followed by a lexical decision task where the answers for some of the previously unrecalled items were presented amongst other words. In some of the experiments, a lexical decision task followed every single definition trial or a small set of definition trials, while in other studies the lexical decision task followed the entire block of definition trials. In some of the studies a second attempt at the recall task occurred sometime after the lexical decision task. The answers to the definition task were primed in the lexical decision task even if the word was not successfully recalled in the definition task, but this priming only occurred when the lexical decision task followed each definition trial or a small set of definition trials. Priming was not observed when the lexical decision task followed an entire block of definition trials so the priming effect was short-term. The presence of priming in the lexical decision task interacted with the feeling of knowing judgments such that priming was only observed for words that generated high FOK ratings. The authors argue that their findings could be explained by a maintained level of spreading activation. However, there have been studies which demonstrate that the presence of priming in the high FOK situations is because people base their FOK rating on how familiar they are with the topics involved with a particular word definition, and people are faster to respond to words where their familiarity with the topic is high (Connor, Balota, & Neely, 1992). This result along with the fact that only short-term priming was found is evidence against the maintained level of activation explanation.

The finding most relevant to the opportunistic assimilation hypothesis is that participants were biased to acquire the answer from the lexical decision task for the trials where they were unable to recall the word. This finding also interacted with FOK so that higher FOK ratings were associated with a better chance that participants would subsequently recall the word successfully on their second attempt at the definition task when the word was presented in the lexical decision task. Seifert et al. (1995) provide only a vague description of how their opportunistic assimilation hypothesis would actually function. Essentially they posit that failure indices are stored when an impasse is reached and that these failure indices will be accessed when relevant information is encountered in the environment. However, they do not define what these failure indices are or how they might be stored or accessed at the right moment. It may be that these indices and associated problem information are maintained at a higher level of activation, but their studies do not provide evidence of this activation especially given that there topic familiarity is an alternative explanation of their priming results (Connor et al., 1992).

Patalano and Seifert (1997) present evidence for a predictive encoding model which posits that in the case of remembering to do some action at a later time participants are more likely to remember to perform the action if they encode the action in terms of the availability of some specific object that would enable the completion of the action. Their task is more associated with prospective memory than problem solving, but their results could indicate that information is more likely to be noticed if the participant has a well developed representation of the problem including cues to the missing elements needed to complete the problem.

The opportunistic assimilation hypothesis was offered as an explanation of insight from the prepared mind perspective of insight problem solving. The prepared mind perspective posits four stages of insight problem solving: preparation, incubation, illumination, and verification (Wallas, 1926). The opportunistic assimilation hypothesis proposes to explain insight by explaining the storage of failure indices in the preparation stage and the noticing of new information in the incubation and illumination stages (Seifert et al., 1995). However, this proposal is far from a theory of the processes that lead to insight. No details are offered on what failure indices are and how they would be accessed when someone is engaged in a completely unrelated task. The question of how information gets noticed and assimilated into problem solving is left unanswered. The authors also offer no direct evidence of such a mechanism operating during problem-solving tasks.

One of the goals of this dissertation is to provide evidence for the hypothesis that open goals direct the acquisition of information relevant to the open goal. The work presented here is consistent with the ideas and evidence presented by Seifert and colleagues (1995), but at the same time there are important differences. One of these differences is that the work presented here provides the beginnings of a mechanistic theory of how open goals are generated and then influence the acquisition and use of problem relevant information. Another difference is that the open goals mechanism proposed here is more general in the sense that it operates across a variety of tasks which are not normally considered problem-solving tasks including prospective memory tasks and attention in dual task situations. The interference of open goals with other tasks is also related to work in the social and clinical psychology literature (Klinger, 1996; Shah & Kruglanski, 2002). The ability to link together findings across a variety of tasks and domains while at the same time showing how open goals operate and explain behavior in problem solving is an indication of the utility and generality of the ideas developed in this dissertation.

### *1.3.2 Open Goals and Insight*

The thesis that is explored in this dissertation is that problem-solving goals that have yet to be fulfilled setup a pattern against which information is matched. As problem solving continues and the person moves on to other goals, information encountered internally or externally that matches the open goal may be incorporated into the process of solving the problem. If this information overcomes a previous impasse or hints at a new type of representation, a feeling of sudden insight may be experienced. Since there was no conscious work going on with the previous goal the insight seems sudden and not to have been preceded by a related train of thought. This hypothesis does not depend on any nonconscious work going on besides the recognition of new information related to an open goal. As will be explained later, this type of nonconscious processing may rely on known processes such as those that lead to priming.

Some of the earliest work on insight problems is consistent with these ideas. For example, Maier (1931) had participants solve an insight problem which involved tying two strings together. The two strings were hung too far apart for the participants to reach one string while holding the other one. The solution to the problem involved the use of an object as a pendulum to set one string in motion and then to catch it while holding the other string. For some participants the experimenter provided an implicit hint by appearing to incidentally set one of the strings in motion while walking by it. Participants who were exposed to this hint, solved the problem by setting the string in motion more often than participants who did not. Furthermore, the majority of participants who solved the problem after receiving this hint and had an insight experience where the solution just came to them did not report any awareness of having received the hint. Making use of this implicit hint is exactly what one would expect if having the open problem-solving goal made participants more likely to incorporate information from the environment into their problem solving.

There have also been more recent studies which have shown that unreportable processing of solution-related information contributes to the experience of insight (Bowden, 1997; Bowden & Jung-Beeman, 2003a). These studies used anagrams and remote associates problems along with subjective ratings of insight experience to examine the manner in which the feeling of insight relates to subthreshold activation of information and unreportable hints.

In insight problems, the initial representation that the problem solver adopts is inappropriate in that it makes the problem difficult if not impossible to solve. The problem solver must re-represent the problem in order to make any progress. In their work with the mutilated checkerboard problem, Kaplan and Simon (1990) sought to understand how a new representation for a problem was found. This problem required that the participants disregard their initial representation and incorporate the concept of parity into a new representation in order to prove that the problem is impossible. A number of different experimental conditions were used in order to determine if the salience of the parity cue would impact the re-representation. They found that the more salient cues led to faster re-representation and proof of impossibility, and they also found a way to extend Newell and Simon's (1972) general theory of problem solving to include re-representation of the problem space during insight problem solving. The basic idea is that there are two problem spaces, and that search for a new problem space is just search through a space of possible problem spaces. Just as there are heuristics to guide people through a normal search space, there should be heuristics and constraints that guide search for a new problem space. Kaplan and Simon identified a few types of these constraints which are divided between external and internal search constraints. External constraints included the cue salience and hints provided by the experimenter, while internal constraints included relevant domain knowledge and heuristics. One of the most common heuristics was the "notice invariants" heuristic. This heuristic focuses the search for a new representation on elements of the problem that remain invariant. Participants who solved the problem faster generated more invariants and noticed perceptual invariants much sooner. Another finding was that hints were most effective in the cases where participants had reached an impasse and were searching for a new representation. This finding is consistent with the idea that reaching an impasse on a problem is necessary for the acquisition of information that may aid in finding a new way to represent the problem.

The idea that failure drives the search for new information is a common finding in both insight and non-insight problem solving (Lovett & Schunn, 1999; MacGregor, Ormerod, & Chronicle, 2001; Ormerod, MacGregor, & Chronicle, 2002). This research provides some insight into when representation change processes are engaged, which is basically when there is a lack of success with the current representation. Lack of success within the current representation seems to be almost synonymous with reaching an impasse. The one difference would be that impasses are usually associated with being unable to generate a new idea for how to proceed, while lack of success could induce a representation change without encountering an impasse. This distinction between impasse driven representation change and failure driven representation change is not one that is usually made in the insight literature. However, such a distinction may be relevant in examining how open goals influence problem solving.

For open goals to successfully influence information acquisition it may be necessary to have a well developed representation of the problem. Reaching an impasse usually requires exploration of one or more representations of the problem. Lack of success in general may also mean that multiple problem representations have been explored even though an impasse has not been encountered. While some studies of insight problem solving relevant to open goals have made the distinction between impasses and non-impasses (Christensen & Schunn, 2005; Patalano & Seifert, 1994; Seifert et al., 1995; Seifert & Patalano, 1991), none of these studies has attempted to dissociate the development and exploration of problem representations from the impasse/non-impasse distinction. Verbal protocols were used in one of the studies presented in this dissertation as an attempt to examine this distinction and how it relates to the open goal hypothesis.

It has been shown that participants can acquire information relevant to insight problems during another task interleaved with their problem-solving activity (Christensen & Schunn, 2005). In this study participants solved some insight problems while occasionally being interrupted to rate the difficulty of other insight problems which were presented along with their answers. Some of the problems from the rating task were analogically related to problems which the participants had left unsolved. Participants often returned to these related problems after the second task and solved them. Participants solved more of the problems for which analogs were presented than those for which no analog was presented. However, participants did indicate awareness of the relationship so they could have been strategically searching for related problems after they initially noticed the relationship. This study provides some support for the idea that useful information can be acquired from other activities. However, in this case both tasks involved solving very similar problems, and the question still remains if information from more distant tasks will be noticed or incorporated into problem solving.

Insight problems can be solved without experiencing a feeling of insight (Weisberg, 1995). The reason for the experience of insight is usually attributed to the suddenness of the solution appearing in one's mind. One explanation of the feeling of insight from an information-processing perspective has been offered by Ohlsson (1992):

> The variable of interest is the complexity of the process required to solve the problem from the state the problem-solver is in immediately after he or she breaks out of the impasse. If the process required to take the problem-solver from this state to the goal state is too complicated to fit within the capacity limitation of human cognition, only

partial insight will occur; if it is simple enough to fit within that limitation, then the problem-solver will experience full insight. (p. 19)

This explanation for the feeling of insight would mean that newly acquired information would elicit a feeling of insight in the case that the new information brought the problem solver directly to the solution or within a limited number of steps of the solution.

These findings in the insight literature provide some support for the idea that open goals may direct the acquisition of problem relevant information. They also indicate that the most likely time for this kind of mechanism to have an effect is when people have experienced failure with their current representation and are searching for a new one. At the same time, an understanding of how open goals influence cognition should further our understanding of insight by helping to understand the processes involved in some forms of representation change.

## 1.4 Incubation

Incubation is one of the stages of insight as laid out by the prepared mind perspective. Some amount of initial problem solving has occurred, but the problem solver has yet to find a solution. More generally, incubation is defined as some break from the problem may that aids problem solving. In the past attempts to find incubation effects have resulted in both success and failure (e.g., Dominowski & Jenrick, 1972; Dreistadt, 1969; Fulgosi & Guilford, 1968; Olton & Johnson, 1976). One review of incubation effects called into question the existence of incubation (Olton, 1979). More recently, reviews of the incubation literature have concluded that incubation is a real phenomena (Dodds, Ward, & Smith, in press; Kaplan, 1989).

There have been a number of theories of incubation including those that involve unconscious work (e.g., Campbell, 1960), forgetting (e.g., Simon, 1966; Smith, 1995b; Smith &

Blankenship, 1991), priming (e.g., Seifert et al., 1995), and combinations of forgetting and priming (Kaplan, 1989). Incubation theories have been divided into two groups: autonomous theories that rely on the passage of time or work within the individual and interactive theories that involved interaction between the individual and the environment (Dorfman, Shames, & Kihlstrom, 1996).

An influence of open goals might seem to imply that incubation is necessarily interactive with the environment providing stimuli which result in insight or ideas that lead to the fulfillment of the open goal. However, problem relevant information may also come from memory. The current representation of the problem may not lead to the retrieval of all relevant information, but as the problem solver leaves the problem behind and moves on to other tasks relevant information may be retrieved and later become assimilated into problem solving. While the environment may have provided the context in which the new information is recalled, the effect on problem solving is indirect as retrieval from the problem solver's associative memory is involved instead of just a direct link between problem relevant information in the environment and the problem itself.

### *1.4.1 Remote Associates and Incubation*

The studies reported in this dissertation use compound remote associates problems to study the influence of open goals. The Remote Associates Test was originally developed as a test of individual differences in creativity by S. A. Mednick (1962). Each problem consists of three words and the task is to generate a fourth word that relates in some way to each of the three words in the problem. This kind of problem has been subsequently refined into compound remote associates problems where the relationship between the answer and the words in the problem is specified such that the answer has to form a compound word or common phrase with each of the other words (Bowden & Jung-Beeman, 2003b). There have been a number of studies which have examined incubation with these types of problems, but these studies have produced mixed results.

Priming answers to remote associates problems was found to produce a marginally significant incubation effect (M. T. Mednick, Mednick, & Mednick, 1964). However, in an experiment designed to test whether incubation effects can be explained by environmental cues Dodds, Smith, and Ward (2002) presented implicit hints to participants during an incubation period between attempts at solving RAT problems. They asked participants to rearrange letters in a given word to form other words. The words given were sometimes answers to the RAT problems participants had seen previously. They found that participants showed the most improvement on the RAT problems when they were shown the answers during the incubation task, but that significant improvement was only found in cases where the participants received some sort of instructions that indicated there was a relationship between the two tasks. The lack of a significant difference when participants were not given instructions indicates that there is a limit on how problem relevant information can be acquired. The lack of effect could be due to the fact that the task in which hints were presented involved mainly orthographic processing and not semantic processing. It has been shown that semantic priming effects are enhanced when tasks involve semantic processing as opposed to a lexical decision task (Becker, Moscovitch, Behrmann, & Joordens, 1997). The lack of an effect in the above study is potentially relevant for the details of the proposed open goal mechanism which influences the acquisition of problem relevant information.

There have also been attempts to study incubation and insight using less direct measures. Instead of looking at improvement on problems after a break, some researchers have examined measures of accessibility such as lexical decision times in order to figure out what is going on during incubation and insight. Answers to unsolved RAT problems are primed relative to neutral words in lexical decision tasks and the amount of this priming is related to the insight experience that participants report (Beeman & Bowden, 2000; Bowden & Beeman, 1998; Bowden & Jung-Beeman, 2003a; Dorfman et al., 1996; Kihlstrom, Shames, & Dorfman, 1996; Shames, 1994). The finding that priming occurs for answers to both solved and unsolved RAT problems is interesting since the word has never been retrieved in the case of unsolved problems. Since the words in the RAT task are all remote associates of the answer word it is surprising that any priming was found.

Other work on incubation with RAT problems has examined how an incubation period affects fixation (Smith & Blankenship, 1991; Wiley, 1998). In this work it was shown that RAT problems are difficult in part because the initial attempts at solving the problem rapidly lead to a fixation effect where interference from previous attempts to answer the problem make it harder to think of new potential solutions. These fixation effects can be induced by priming incorrect associations, but they can also occur naturally as participants use their knowledge to generate potential solutions based on associations with the three words making up the RAT problem. In cases where the fixation is induced by priming incorrect associations, an incubation period helps (Smith & Blankenship, 1991), but in cases where fixation was caused by long-term associations like those found in the development of expertise the incubation period was did not successfully reduce fixation (Wiley, 1998).

The idea that open goals influence problem solving is highly relevant to understanding incubation. While some incubation effects may be due to the passage of time, it is likely that other effects involve priming or an interaction of forgetting and priming. The mixed findings of incubation studies with RAT problems may be explained in terms of an interaction of forgetting and priming with appropriate information. These mixed findings may provide useful constraints as they may indicate that there must be some degree of representational overlap between the problem representation and the priming stimulus. These ideas are consistent with the idea of levels of processing and transfer appropriate processing found in the transfer literature (Craik & Lockhart, 1972; Lockhart, Lamon, & Gick, 1988).

### 1.5 Transfer

A specific case of the interactive incubation effect is what has been referred to as reversed analogical transfer (Christensen & Schunn, 2005). While the standard transfer study consists of presentation of a set of source material followed at some later time by a target problem where the source information would be useful, in the reversed transfer case the target problem is attempted before the source material has been presented. Essentially the problem solver works on the problem but does not solve it, and during a break from problem solving the source information is presented before resuming work on the target problem. In this kind of reverse analogical transfer Christensen and Schunn found a much higher rate of noticing relevant analogies than is normally found in analogical transfer studies (e.g., Gick & Holyoak, 1980). A framework for this kind of reversed analogical transfer has also been discussed by Langley and Jones (1988).

The research on analogical transfer has generally found that surface similarities influence which source information is noticed or retrieved, but that structural similarities are key to mapping the source information to the current problem (Forbus, Gentner, & Law, 1995; Holyoak & Koh, 1987). However, the study of analogy in naturalistic environments and in relation to expertise has shown that noticing analogies based on structural similarities may not be as rare as studies of analogical transfer have indicated (Dunbar, 2001; Novick, 1988). The finding that analogy use is not as rare in naturalistic settings is consistent with the notion that open goals influence information acquisition from the environment thereby increasing the chances that relevant analogies would be noticed. Experts are likely to have a deeper conceptual representation of the problem they are working on which may also influence the likelihood that information relevant to an open goal is noticed and assimilated into their problem solving.

People are not always aware that they are using old knowledge in a new problem. Schunn and Dunbar (1996) found that knowledge from a problem solved on one day influenced the likelihood of solving a different type of problem on the second day. When both problems relied on the concept of inhibition there was an increased proportion of participants who solved the second problem than in the condition where the two problems shared no common concept. The authors interpreted this as a form of transfer by priming. Since participants had prior knowledge of inhibition, the first problem served to increase the accessibility of this information so that it was more likely to be used on the second day. However, most participants did not report being aware of the connection between the two problems. This result illustrates that people can be facilitated in problem solving while being unaware of the source of the knowledge. When combined with the notion that open goals would increase the chances of acquiring problem relevant knowledge, this could go a long way towards explaining the fact that many accounts of insight seem to be accompanied by the feeling that the relevant information just appeared in one's mind.

### 1.6 Creativity

The Remote Associates Test was originally developed to measure individual differences in creativity (S. A. Mednick, 1962). The idea behind the RAT was that associative processes were the key processes in creative ability, and the RAT would be able to measure individual differences in these processes. While the RAT was a reliable measure and was found to correlate with other measures of associative behavior (S. A. Mednick, 1962), attempts to related RAT scores to creativity in real-world domains were not very successful (e.g., Andrews, 1975). While RAT problems are used in the studies presented in this dissertation, they are not being used as a measure of creativity. They are useful because they are short insight-like problems, and a recent set of norms have been published for RAT problems presented for varying amounts of time (Bowden & Jung-Beeman, 2003b).

The idea that open goals influence the acquisition and use of problem relevant information is related to research on creativity in that it provides one possible explanation for where creative ideas come from. If open goals influence which information gets incorporated into problem solving, then having open goals should increase the chance that some distantly related information gets noticed as the problem solver sets aside the problem and works on seemingly unrelated tasks. The notion that creativity arises from unlikely combinations of ideas is prevalent in the creativity literature.

For example, it has been proposed that creative ideas arise from a blind variation mechanism which produces combinations of ideas which are selectively retained (Campbell, 1960; Simonton, 1999). These theories rely on a fair amount of nonconscious processing to occur as different elements are combined before being brought to mind when they are retained by some unspecified nonconscious process. This kind of idea is not incompatible with an open goal mechanism that filters incoming information and compares or combines it with an open goal. An open goal mechanism could produce the same results being proposed in these variation and selection models of creativity while explaining exactly what kind of processing is being done outside of awareness. The kinds of processes being posited as part of the open goal mechanism are processes of attention, perception, and memory retrieval that are already known to go on mostly without awareness.

### 1.7 Role of this Mechanism in Problem Solving

The problem-solving research that has examined how information relevant to unsolved problems is acquired has focused on the role such a mechanism may have in overcoming impasses in insight problem solving (Christensen & Schunn, 2005; Dodds et al., 2002). While a theory that focused on insights and incubation effects would be interesting, a more general mechanism could explain a number of problem-solving phenomena in addition to insights. In addition, linking this mechanism to other areas of cognition leads to a more powerful and parsimonious theory.

Insight problems are a particular class of problems in which people often experience a sudden insight or “Aha!” experience. These problems can be either well structured or ill structured, but often times researchers study these problems as a way to understand how people achieve insights in real world problems. Real world problems are generally ill structured and more complex than problems usually studied in the psychology experiments. Ill structured problem solving is not understood as well as the well structured problems that have been most studied in the psychology lab. One attempt at conceptualizing some of the similarities and differences between well and ill structured problems is an extension of the Newell and Simon (1972) problem space by Simon (1973). He noted that one of the differences in solving the two problems is that the problem space can change as one solves an ill structured problem, and one of the main ways in which such a problem space can change is through the assimilation of new information from either internal or external sources. This process of noticing new information is not necessarily driven by impasses in problem solving.

Ill structured problems are often complex enough to require a person to decompose them into subproblems preferably in a way such that the subproblems have only minimal interaction (Simon, 1969). This sets up a situation where a number of problem-solving goals can be present in memory, but only one is being worked on at a time. This situation is one where it may be beneficial to have a mechanism that can notice information related to other goals which are not being pursued at the time. For example, in the process of working on one subproblem one may encounter information in the environment or one's own problem solving that solves another subproblem, and so it may be beneficial to switch to that other goal in order to satisfy it. In fact, as described below, there are a few examples of researchers noticing this kind of goal switching behavior in engineers or designers who are working on a problem. The domain of problems in which such a mechanism plays a role probably also includes problems which people abandon or put aside for reasons other than reaching an impasse.

The mechanism can also be generalized by considering other types of representation changes besides insights. Insights are generally considered to involve a representation change or restructuring of the problem representation. However, there are many cases in problem solving where representation change often goes unnoticed. In terms of the problem space model of problem solving, a representation change would occur any time an aspect of the problem space is modified. This could be something as simple as the creation a more efficient move operator or as complex as a change to every aspect of the problem space. In many cases, smaller changes such as a new move operator have been referred to as learning and not as representation change. That distinction does not seem particularly relevant for the theory presented here as they are all changes to the current representation being employed, and these changes may come about through the influence of open goals.

### *1.7.1 Example from Engineering Design*

As an example of problem-solving behavior that could be explained by the influence of open goals on the acquisition of problem relevant information, there have been studies that have found that expert designers show unexplained deviations from their normal structured problem-solving process. The main issue addressed in these studies is the extent to which the design problem-solving process is structured, and if it is structured in what order designers work on subproblems.

Guindon (1990) observed a number of cases where designers deviate from a structured approach to the design process. Deviations occurred when designers discovered some new knowledge that was relevant such as a partial solution to a sub problem, and they immediately shifted attention to that subproblem. When a new requirement was generated, the designers often switched to solution development for that requirement. Similar findings are presented from in a series of protocol studies of engineers (Visser, 1990, 1996). Deviations in problems solving were observed as the designer deviated from his plan to use the same information in two different ways. Another example is when defining aspects of one component leads to defining the same aspects of other components with similar representations. It should be noted that the task used in this study is a highly structured one for a design task, but the results seem consistent with what has been found in other studies using less constrained tasks. The author claims that the active mental representation of a component may lead to the activation of components with similar representations, and the engineer was observed to transition between such components in different submodules of the design.

Another set of studies investigating engineering design through observations and verbal protocols also support a number of the ideas mentioned so far while providing further details on the process of decomposition (Ball, Evans, & Dennis, 1994; Ball, Evans, Dennis, & Ormerod, 1997; Ball & Ormerod, 1995). The majority of observations are shown to be consistent with a combined top-down breadth-first and depth-first approach. They propose that novices tend to do design in a depth-first approach possibly because this minimizes memory load for goals and incomplete design states, but experts seem to operate in a more breadth-first approach. The protocols they analyze from an engineering design task show that 88% of transitions between design activities were consistent with a breadth-first design process. The main exception to the consistent transitions was when a designer was unsure of a certain conceptual solution to a subproblem and rapidly developed that solution a more detailed level in order to assess its feasibility. Another exception occurred when a designer halted progress on one subproblem in order to work on analogous parts of another subproblem. The solution for one subproblem then guided the solution for the analogous subproblem. The designers also sometimes reached an impasse which they would try to circumvent by generating new ideas or focusing on another subproblem.

These results are suggestive of the role that open goals may play in real-world problem solving in complex domains. The fact that engineers moved to a different part of a problem when encountering an impasse and switched to relevant subproblems when relevant information was encountered can be explained by their noticing the relevance of new information to open goals associated with unsolved problems or subproblems. Expertise may play a role in what information gets noticed as one of the general findings of the expertise literature is that experts tend to represent the deeper structural concepts of a problem in their domain such as the function of a device in engineering design (Moss, Kotovsky, & Cagan, 2006). The representation of a problem is likely to have an effect on what information related to open goals gets noticed as it is hypothesized that incoming information is matched against some kind of pattern or semantic representation associated with the open goal.

## 1.8 Mechanisms

Zeigarnik (1927/1938) originally proposed that interrupted tasks were recalled better than completed ones because of some form of tension which is only relieved by completion of the task. This explanation is vague, but it does point to the possibility that there is some aspect of the cognitive system that maintains these incomplete goals. A variety of explanations have been used to explain the findings described above. These include a maintained level of heightened activation for open goals (Goschke & Kuhl, 1993; Marsh, Hicks, & Brink, 1998; Rothermund, 2003; Yaniv & Meyer, 1987) and an increased facility for forming associations between open goals and relevant information (Yaniv et al., 1995). There has also been an explanation in terms of failure indices placed in long-term memory (Seifert et al., 1995).

Explanation in terms of a maintained heightened level of activation seems to be the easiest to understand in terms of current cognitive architectures such as ACT-R (Anderson et al., 2004; Anderson & Lebiere, 1998). However, there has also been evidence that some of the results attributed to heightened activation may be explained by other factors such as topic familiarity (Connor et al., 1992). The other explanations that have been suggested are less clear. For example, there is no explanation for why it should be easier to form associations when there has been a failed retrieval attempt (Yaniv et al., 1995). The opportunistic assimilation idea supposes there are entities called failure indices that facilitate access to unsolved problems but these entities are not described, and it is not clear how they would facilitate access to unsolved problems (Seifert et al., 1995). Relevant issues include how such indices are created, what they contain, how relevant information is recognized, and what happens after they are activated by relevant information. Answering these questions requires a theory that specifies how problem-solving activity interacts with processes like memory and attention.

One advantage of modeling these tasks with an established cognitive architecture like ACT-R is that it requires the theorist to specify these interactions while at the same time working within the framework of an established theory of memory and attention. One of the final sections of the dissertation examines one such model that fits with the results of the studies presented in this dissertation. The empirical work was designed to explore properties of the proposed open goal mechanism, and the initial modeling work begins to demonstrate how it can be implemented and incorporated into existing theories of problem solving.

Any candidate theory must be able to show how an open problem-solving goal becomes established in such a way that it exerts an influence on information that is acquired. This includes a way of selecting or biasing related information for further processing, and for incorporating this information into the problem. A couple of existing mechanisms may be related to this type of processing. These are spreading activation mechanisms and the idea of semantic attractors in distributed models. Both of these mechanisms have been used to explain phenomena like semantic priming, feelings of knowing, and acquisition of information (Becker et al., 1997; Plaut & Booth, 2000; Yaniv & Meyer, 1987; Yaniv et al., 1995). The essential property of these mechanisms is that they bias processing in favor of concepts that have a greater degree of featural overlap with recently encountered concepts. Showing how these same processes may be used in representation change is one way to build a theory that covers a variety of phenomena while also remaining parsimonious. It is also possible that other mechanisms of incubation that have been proposed such as forgetting or decay of information that is detrimental to problem solving can further enhance the power of the open goal mechanism in aiding problem solving.

### 1.9 Overview of Studies

In general, there is evidence throughout the literature that open goals influence cognition. The studies presented in the following chapters were designed to provide further evidence that open goals do influence cognition and specifically to show that they influence the acquisition and use of problem relevant information. Furthermore, they were designed to explore properties of this open goal mechanism in order to help constrain the development of a theory of how open goal influence cognition. All of the studies make use of design where participants work on a set of problems, followed by a second task where implicit hints are presented for some of the problems that participants left unsolved. After the hints have been presented, the participants are then given the same set of problems to work on again.

RAT problems are used in the proposed work due to the fact that they are short insight-like problems where the knowledge level of a participant is not a concern. Since these problems are solved primarily by memory retrieval they should be useful in studying relative activation levels in memory as maintained activation is one of the primary explanations used to explain the effects of open goals. The RAT problems used in each of these studies can be found in Appendix I.

Studies 1A and 1B demonstrate that it is possible to see an effect of open goals in this kind of experimental paradigm. Study 1B shows that the results of Study 1A can not be explained by assuming that the hint is just priming the answer. Study 2 examines whether the number of open goals has an impact on hint effectiveness. Study 3 manipulates the time that participants have to work on the RAT problems while also having participants think aloud as they solve the problems in an effort to assess whether reaching an impasse interacts with hint effectiveness. This protocol study also allows for an assessment of the degree of fixation that occurs between problem attempts.

# Chapter 2

# Initial Studies

## 2.1 Study 1A

The first study is an extension of previous work which serves to replicate findings concerning acquisition of information as well as reported priming effects with RAT problems. While RAT problems have been shown to induce priming in answer words regardless of whether they were solved or not, the task in which priming has been measured always immediately followed each RAT problem. This study looks for a longer lasting priming effect. If a maintained level of activation is associated with the way open goals influence cognition in real-world problems involving long periods of time, then longer-term priming should be observed.

Failed recall attempts have been shown to bias acquisition of information in other tasks (Yaniv & Meyer, 1987; Yaniv et al., 1995), but the same thing has not been demonstrated for a problem-solving task. Actually, the opposite result has been shown whereby the answer is acquired only if participants are explicitly informed about the relationship of the two tasks (Dodds et al., 2002). However, their study used a task which emphasized orthographic processing and not semantic processing. The current study uses a lexical decision task to present the answers which should allow for more semantic processing.

### *2.1.1 Method*

*Participants*. The participants were 39 undergraduate students at Carnegie Mellon University who completed the study as part of a course requirement. All of the participants were native English speakers.

*Materials.* The remote associate problems used in this study are similar to those appearing on the Remote Associates Test (RAT) developed by Mednick (1962), but they were taken from a recent set of normed RAT items (Bowden & Jung-Beeman, 2003b). The 20 RAT problems we used were chosen from the normed set so that the mean proportion of participants solving them was .51 with a range of .38-.64. For each answer to the RAT problems, three words of the same length and similar frequency were selected from a database of words (Balota et al., 2002) to serve as control words. A set of 20 nonwords were also obtained from the same database. In addition, there were two practice RAT problems, 15 practice words and 15 practice nonwords. The set of words and RAT problems were generated so that none of the words were associated with any of the RAT problems, and none of the words in the RAT problems were associated with words in another RAT problem. Word association was determined from a set of word association norms (Nelson, McEvoy, & Schreiber, 1998).

*Procedure.* The experiment involved two tasks which consisted of two blocks of RAT problems and an intervening lexical decision task. All tasks were presented on a computer with a 17-in. monitor using E-Prime software(Schneider, Eschman, & Zuccolotto, 2002). The basic design of the study was a set of 20 RAT problems followed by a lexical decision task where answers to some of the RAT problems appeared as stimuli. The lexical decision task was followed by another attempt at the same set of RAT problems. The two factors in the study were whether the RAT problems were initially answered correctly or incorrectly and whether or not the answer was presented as an implicit hint during the lexical decision task.

Participants were given instructions on how to complete both tasks, and they were told that the experiment would involve alternating between the two tasks a few times. The RAT problems were called word association problems, and participants were told that they should answer the problems by generating a word that forms a word or common phrases with each of the other three words. They were told that they would be given 30 seconds to answer each problem. There were only allowed to enter one attempted answer for the problem. In order to discourage frequent guessing in order to finish the experiment faster, a score was presented on the screen after each RAT problem in addition to feedback about whether the problem was answered correctly or not. The score was increased by five points for each correct answer, decreased by two points for each incorrect answer, and did not change when no answer was provided. The RAT problems were presented as three words arranged in a vertical column in the center of the screen with an outlined box presented beneath the last word in which the participants typed their answer. After the problem was answered, the current score was presented for 1 s followed by the next problem. Participants were given two practice RAT problems followed by 20 practice lexical decision trials. This was followed by the presentation of the 20 RAT problems.

After completing the 20 RAT problems, the participants completed a lexical decision task consisting of 25 words and 25 nonwords. Each trial began with a fixation cross presented in the center of the screen for 1500 ms followed by the word or nonword which remained on the screen until the participant responded. A blank screen was presented for 500 ms between trials. The first 10 trials were considered practice trials and consisted of 5 words and 5 nonwords presented in a random order for each participant. The remaining 20 words consisted of 10 control words and 10 words that were answers to the previously presented RAT problems. When possible, the answer words consisted of 5 answers to RAT problems that the participant got correct and 5 words from problems that the participant got incorrect or failed to answer in time. A control word was chosen for each answer word that was presented so that the control was of the same length and a similar frequency as the answer word. This was accomplished by randomly choosing one control word from a list of three words that had been assembled matched on frequency and length for each RAT answer. This was done to control for possibility that the RAT problems that were not answered had answers that differed in some way from correctly answered problems that would differentially affect reaction times for the two classes of problems. For example, the unanswered problems may have had answers that were of a lower frequency than the correctly answered problems. These 40 words and nonwords were presented in random order.

In cases where a participant did not have at least five incorrect or five correct answers, the number of lexical decision trials was necessarily reduced by one control word and one answer word for each number less than five. For example, if a participant only answered three problems incorrectly, there would only be 8 control words and 8 answer words included in the lexical decision task.

After the lexical decision task was completed participants were informed that they would now be presented with the same set of 20 RAT problems again. This was the first time the participants were informed they would see the same problems again. After completing this final set of RAT problems the participants were debriefed.

### *2.1.2 Results*

Data from three participants were excluded because they guessed at the answers to the RAT problems in an apparent attempt to finish the study quickly. The criteria for exclusion were answering more than 15 RAT problems in either of the two sets of 20 and getting less than 50% of the answered problems correct.

The lexical decision response times were analyzed in order to determine if there was any evidence of priming for answers to the RAT problems relative to the control words (see Figure 1). The relevant factors are whether the RAT problem corresponding to the presented hint word was solved or not and whether the word presented was a hint (i.e., an answer to a RAT problem) or was a control word. A 2 x 2 within-subjects ANOVA was conducted on the mean correct lexical decision reaction times for each participant. There was no significant main effect of whether the RAT problem was answered correctly, but there was a marginally significant priming effect, $F(1, 35) = 2.99$, $p = .09$. The interaction was not significant. This result indicates that there was a tendency for the answers to both solved and unsolved RAT problems to be primed relative to the neutral control words.

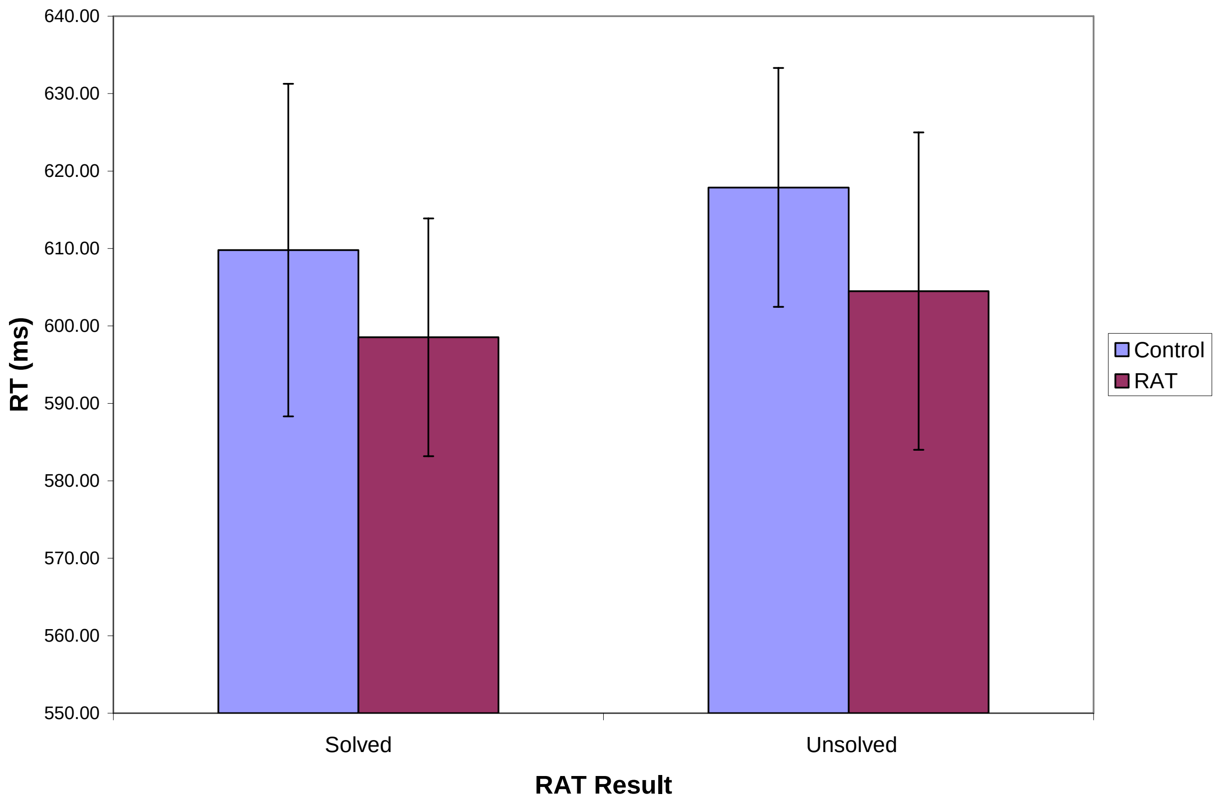


Figure 1. Response times for the lexical decision task

The main issue in the study was whether the hint embedded in the lexical decision task was effective at increasing the proportion of problems solved. The improvement from the first attempt at the RAT problems to the second attempt was assessed by examining the proportion of RAT problems that were solved correctly on the second attempt after being left unsolved initially (see Figure 2). For this analysis, three additional subjects were excluded for having fewer than two RAT problems that were initially unsolved and assigned to the no hint condition. This occurred because the program that ran the experiment randomly chose 5 unsolved problems to include in the hint condition, and this method sometimes left too few unsolved problems in the no hint condition. A Wilcoxon Signed Ranks Test was run since the data were highly skewed, and there was a significant difference between the proportion of previously unsolved problems that were improved upon in the hint and no hint conditions, Z = -2.27, p = .02. Participants improved more on problems for which a hint was presented which means that the hint was effectively acquired and used in problem solving some of the time.

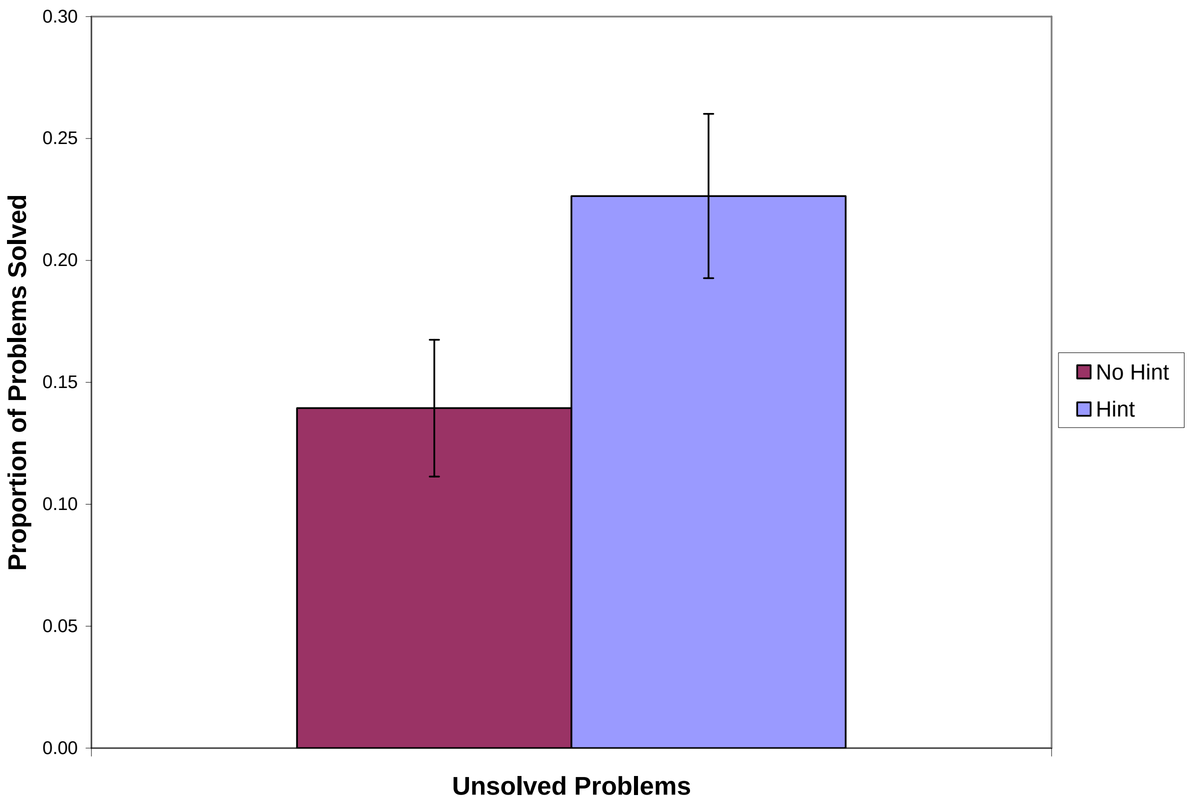


Figure 2. Proportion of unsolved problems solved on second attempt

The success rate for the initial block of RAT problems was also examined in relation to improvement on the second block of RAT problems in order to assess the plausibility of the hypothesis that having fewer open goals would lead to a higher chance of improvement on problems where the answer was presented. There was a significant correlation between the proportion of  problems solved on the initial block of RAT problems and the amount of improvement on the unsolved problems for which a hint had been presented, $r = .36$, $p = .03$, but there was no significant correlation between initial success and improvement for problems in the no hint condition.

### *2.1.3 Discussion*

The results demonstrate that the hints presented in the lexical decision task did improve performance on the RAT problems more than just a second attempt at the same problems. Participants were not explicitly informed of the relationship between the tasks, and so the results seem to contradict the findings of Dodds et al. (2002). However, it could be that the tasks used to present the answers differed in that the lexical decision task may have required some semantic processing which the word finding task used by Dodds et al. did not. This issue is explored in Study 2.

The marginally significant priming effect of the first block of RAT problems on the hints in the lexical decision task is interesting because it replicates similar findings from other work on memory retrieval and RAT problem solving, but the priming was found over many more intervening problems and over a longer period of time than in prior work. This result provides some support for the idea that activation is being maintained over a relatively longer period of time.

The correlation between initial success and improvement on problems for which hints were presented may be indicative of a limit on the number of open goals that can successfully be maintained. As the number of open goals increases, the ability to recognize relevant information may decrease due to interference or a resource limitation. Another possible interpretation of this correlation is that participants who performed better on the RAT problems were also better at making use of the hints. The issue of the number of open goals is addressed directly in Study 2.

### 2.2 Study 1B

The purpose of this experiment is to compare the effectiveness of presenting problem relevant information before the problem had been attempted to the effectiveness of presenting the hint after an open goal had been established. This situation is similar to comparing the standard transfer case of presenting source information followed by a target problem to the reversed case of working on a target problem before seeing the source information (Christensen & Schunn, 2005).

In terms of Study 1A, this study evaluates what effect presenting the answer to a RAT problem has on problems that have yet to be attempted while also attempting to replicate the results of Study 1A. In the first study, participants improved more on RAT problems when the answer was presented during a separate lexical decision task even though they were not informed about the relationship between the two tasks. One explanation for this result that has nothing to do with open goals is that the words in the lexical decision task increased the activation of the answers so that participants were able to retrieve the correct answers when shown the RAT problems for a second time. This experiment evaluated the effect of presenting the answer for the problem before seeing the problem so that the effect of increased activation for the answer word can be evaluated without the presence of the associated open goal.

In comparing the effectiveness of the hint in these two conditions, one issue that must be considered is that participants can experience fixation on RAT problems due to interference from the first few words or concepts that are recalled (Smith & Blankenship, 1991). The idea is that it becomes more difficult to think of new associated words due to interference from the words already recalled. Smith and Blankenship (1991) demonstrated that some of this interference can be overcome by taking a break from the problem so that the previously thought of associates decay with time making it easier to think of new items.

Given that interference plays a role in the difficulty of solving these problems it should be expected that presenting the answer word before there has been any attempt at the problem would be expected to have a larger impact on RAT problem solving than in the case where there has already been a failed attempt at problem solving. If it is the case that presenting the word just raises its activation level which increases its probability of recall, then having no prior interference from distractor words would increase the effectiveness of presenting a hint relative to the case where there are other activated words competing for recall.

### *2.2.1 Method*

*Participants.* The participants were 31 people from the Carnegie Mellon community and surrounding areas who participated in exchange for $6.

*Design and procedure.* The procedure was similar to that of the first experiment with three changes. This study employed a blocked design where blocks of 5-8 problems are presented with lexical decision tasks in between each block. There were four pairs of blocks with each pair being made up of one block where RAT problems were initially presented and the second block which involved a second presentation of the RAT problems from the first block as well as previously unseen problems. In between the two blocks making up the pair a lexical decision task was given where answers to half of the unsolved problems from the preceding block were given. In addition answers to half of the previously unseen problems were given where the unseen problems are new problems that are mixed in with the second presentation of the original problems in the block immediately following the lexical decision task. The first and second presentations of problems are separated by one lexical decision task where some of the answers to unsolved and unseen problems are presented, and this design was chosen in the hopes that it would increase the effect of presenting the answer since participants would have at most five unsolved problems in a block before doing the lexical decision task. The participants were not informed that any of the problems would be repeated.

There were a total of 30 RAT problems selected from the norms published by Bowden and Jung-Beeman (2003b). The same constraints used in Study 1A to limit relationships among different problems and their answers were used in selecting the 30 problems. In addition, the problems were selected so that the expected success rate for 30 seconds of problem solving was 50%. For each participant 20 of these problems were randomly selected to be presented twice, and the remaining 10 were presented as previously unseen problems in the second block of each pair of blocks. The 20 problems were broken into four blocks of five problems each, and these blocks were the first member of each pair of blocks. The second pair of each block consisted of five problems that had already been seen and 2-3 problems that were previously unseen, with half of these unseen problems having their answer presented during the lexical decision task separating the two blocks of each pair (see Figure 3 for an illustration of the design). Overall, five of the ten unseen problems had their answers presented before being attempted. Since the expected success rate was 50% based on the norms for these RAT problems (Bowden & Jung-Beeman, 2003b), it was expected that 10 of the original 20 problems would be left unsolved after 30 seconds of work and half of these 10 would have their answer presented before the second attempt at solving them.

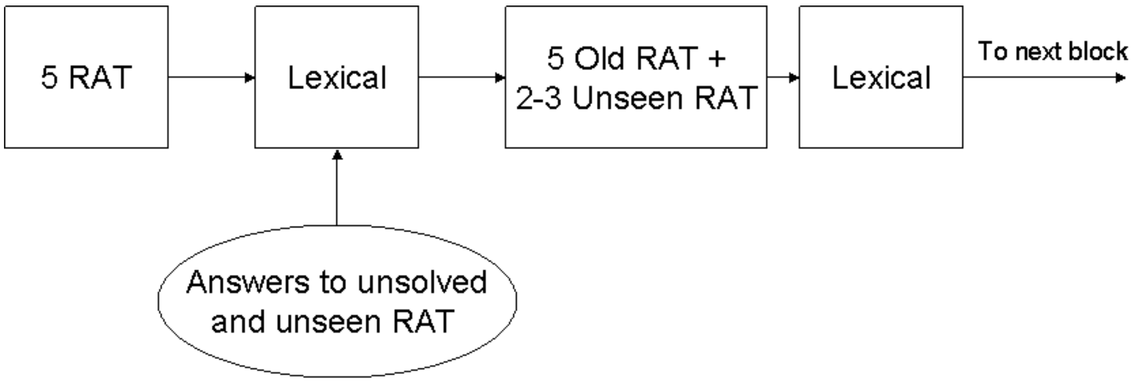


Figure 3. Design of Study 1B

Each of the eight blocks of RAT problems was separated by a lexical decision task, but only the lexical decision tasks between pairs of corresponding RAT blocks were related to the RAT problems. Each lexical decision task consisted of an equal number of words and pronounceable nonwords. An effort was made to make each lexical decision task contain about 20 items, but there was some variation in this number because half of the problems left unsolved had their answers presented in the lexical decision task. The blocks that did not separate pairs of corresponding RAT blocks each consisted of 10 words and 10 nonwords all of which were unrelated to the RAT problems. The structure of the remaining lexical decision blocks was more complicated because the number of hint words needed varied some from block to block. The lexical decision blocks which occurred between pairs of corresponding RAT blocks consisted of 9 words and 9 nonwords plus the answers to unsolved and unseen problems and an equal number of nonwords.

For example the average participant might leave 2 of the 5 RAT problems in a block unsolved. One of these two problems would have its answer presented along with an additional nonword. In addition there would be an additional answer for an as of yet unseen problem presented in the lexical decision task along with an extra nonword. This would bring the total number of trials in the lexical decision block to 22 trials.

A pool of 107 filler words unrelated to the problems and their answers and a pool of 168 nonwords were used in constructing each lexical decision block. The order of trials within a block of lexical decision trials was randomized.

The participants were instructed on how to perform both tasks in the study. For the RAT problems they were told that their task was to generate a fourth word that could be combined with each of the three given to form a compound word or common phrase. Each problem was presented with the three words arranged vertically near the center of the screen. A box with a cursor was placed beneath the last word where participants could type their answer. If the participant entered an incorrect answer, the computer made an error sound and the box was cleared. The participants still had whatever was remaining of the 30 seconds for each problem to continue working on the problem. If the problem remained unsolved after 30 seconds, the problem was cleared and the word “Unsolved” was presented in the center of the screen for two seconds. If the correct answer was entered the problem was immediately cleared from the screen and the word “Solved” was presented for two seconds. The next RAT problem was presented immediately after the feedback from the previous problem.

The participants were instructed on which key to press for the word and nonwords response. Each lexical decision trial began with a fixation cross displayed for 1.5 seconds and ended when the participant responded. A blank screen was then presented for 500ms followed by the fixation cross for the next trial. The first six trials of each lexical block consisted of three words and three nonwords which were considered warm-up trials. None of the answers were presented during these trials, and they were not included in any of the analyses. These trials were included to allow the participant to fully switch to the lexical decision task before collecting reaction times for the critical experimental stimuli.

After the participants had finished the eight blocks of RAT problems they were given a surprise free recall task in which they were asked to recall as many of the RAT problems as possible. They were told that order was not important and to write down as much of a RAT problem as they could remember even if it was only one of the three words given. The participants were allowed as much time as they wanted to recall the problems. After the recall task, they answered a questionnaire to assess their awareness of the relationship between the RAT and lexical decision tasks. The questionnaire also asked about their strategies for solving the RAT problems and whether they had tried to rehearse or work on the problems between presentations of the problem.

### *2.2.2 Results*

One participant was excluded from all analyses because more than 70% of his lexical decision response times were greater than one second.

#### *Lexical Decision*

Accuracy on the lexical decision task was greater than 95%. In order to determine if there was a priming effect in the lexical decision task, the mean response times for the words that were answers to previously unsolved RAT problems were compared to the mean response times for the yet to be seen RAT problems. Only trials for which participants responded correctly were included. This data is presented in Figure 4. There was no significant difference in the response times to these two types of words. This result indicates that there was no reliable priming effect for answers to unsolved problems relative to neutral words.

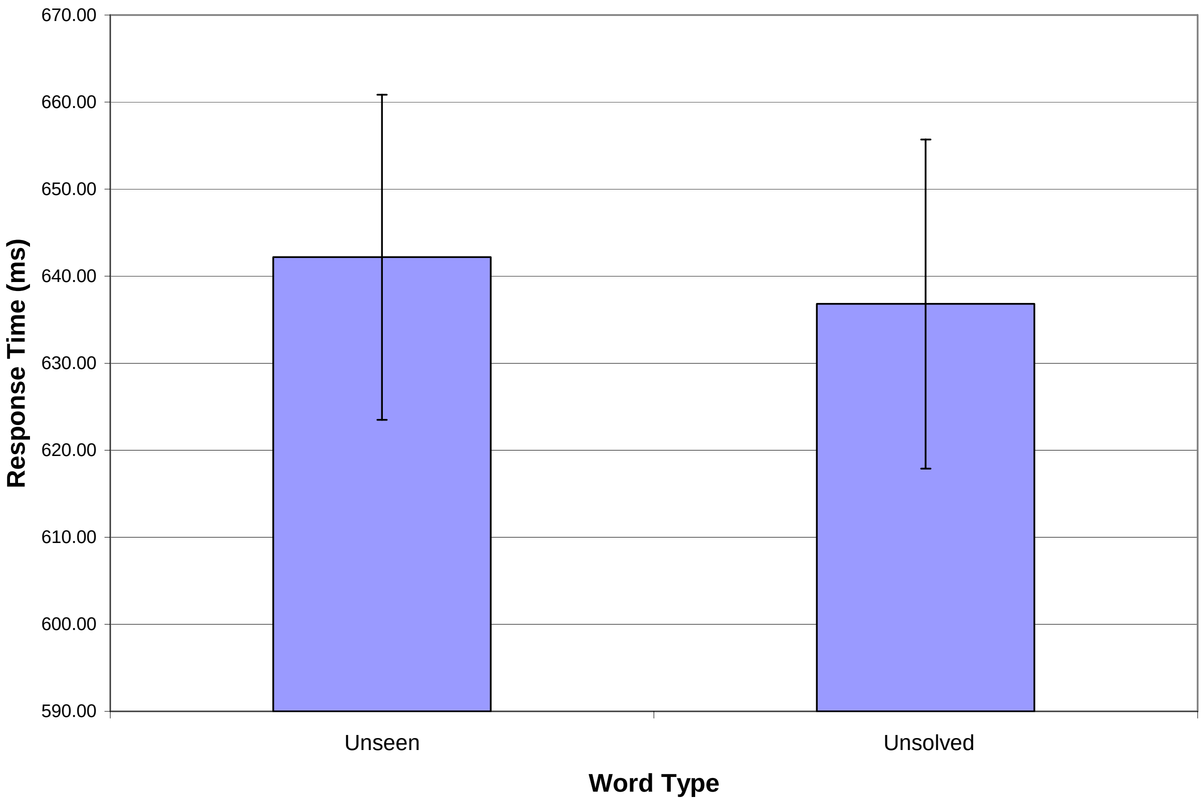


Figure 4. Response times by word type for lexical decision task

*Improvement on RAT Problems*

The design of the study includes the initial presentation of the RAT problems that will be presented a second time and the presentation of previously unseen problems mixed in with the second presentation of the RAT problems. It was expected that there would be no difference between the initial presentation of the RAT problems and the unseen RAT problems which did not have hints presented since these are both cases where the problem is seen for the first time without first seeing a hint. These two conditions did not differ significantly, $t(29) = 1.39$, p = .18, and so the problems from these conditions were combined into one condition which consisted of all RAT problems where no hint had been presented before the initial presentation of the problem.

The proportion of problems solved in each condition can be seen in Figure 5. The data were analyzed with a 2 x 2 within subjects ANOVA with the factors being whether this was the first or second attempt at the problem and whether a hint had been presented. There was a main effect of whether the problem had already been attempted, $F(1,29) = 62.32$, $p < .001$, and a main effect of the hint, $F(1,29) = 5.91$, $p = .02$. There was no interaction ($F < 1$). In terms of the difference in the proportion of problems solved, the presentation of the hint in the lexical decision task had an equal effect on both unseen and unsolved problems. The fact that presenting the hint produces an increase of the same magnitude for both unsolved and unseen problems is evidence for some effect of open goals since problems which had already been attempted were more difficult (i.e., they were the unsolved subset) and should have a higher interference level from other incorrect answers which had already been examined and rejected by the participant. Another way to look at this data which takes into account the relative difficulty of the unsolved problems is that seeing a hint led to a 48% relative increase in the proportion of unsolved problems that were solved on the second attempt while there was only a 22% relative increase in the proportion of problems solved in the case where the problem was seen for the first time. As an additional note, the problems which participants had answered correctly the first time were answered correctly on the second attempt 98% of the time.

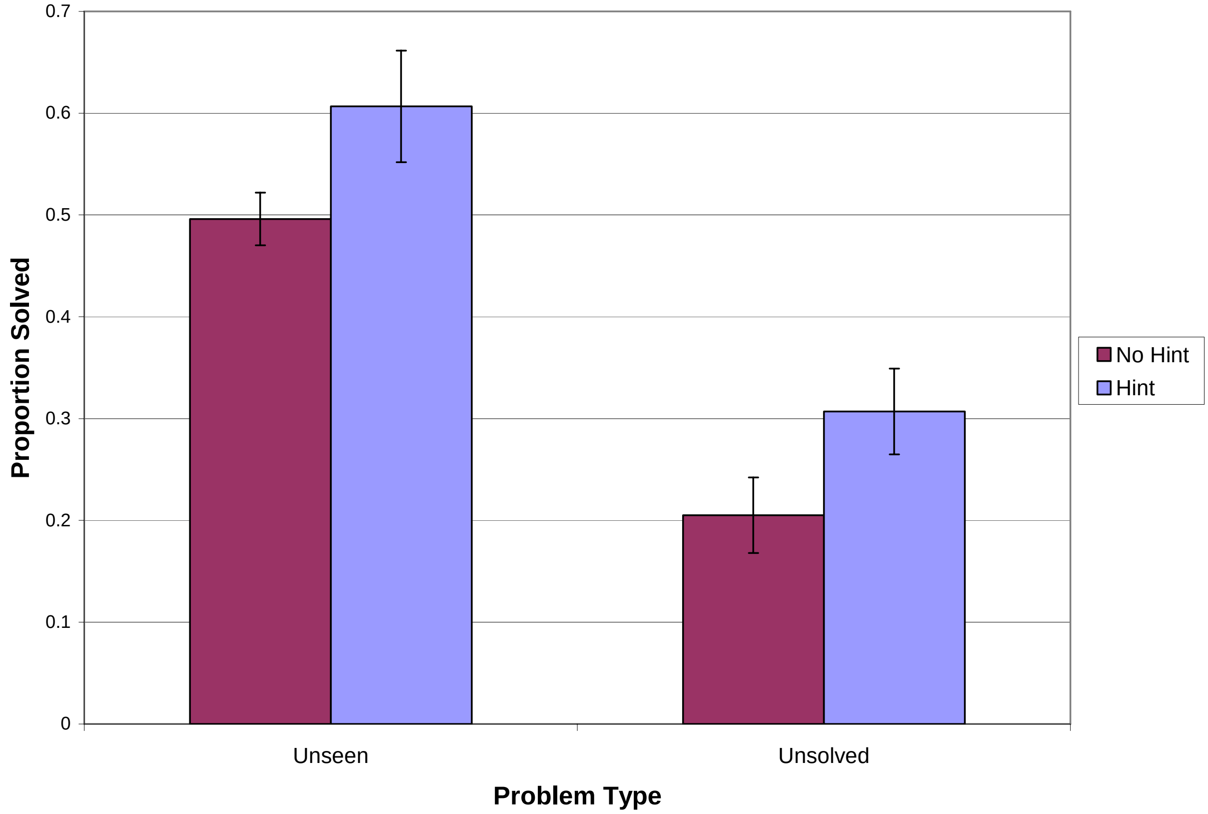


Figure 5. Percent of problems solved in each condition

Solution time for solved problems was also examined. The question of interest is that given that a participant solved a certain number of problems in a given condition, how long did it take to solve those problems on average. Not all participants solved a problem in every condition, and so if a participant did not have any solution time for a condition then they were given a solution time of 30 s for that condition since 30 s was the time limit used in this study. Since the initial attempt at RAT problems did not differ from the unseen problems where no answer was presented, $t(29) = 1.35$, $p = .19$, they were again combined. The solution time data are presented in Figure 6. The data were analyzed with a 2 x 2 within subjects ANOVA. Of the 120 data points in the analysis, 19 were points for which 30 seconds was assigned because of a participant who failed to solve a problem in a particular condition. There was a main effect of attempting the problem before, $F(1,29) = 29.07$, $p < .001$, and a main effect of presenting the answer, $F(1,29) = 4.27$, $p = .048$. There was also a significant interaction, $F(1,29) = 8.26$, $p = .008$, which was due to the fact that the hint had an effect on solution time when the problem had previously been left unsolved but not when the answer was presented before the problem was attempted. The lack of a difference in solution time for the unseen problems was not likely due to a floor effect as many problems were solved faster than 10 seconds, and it appears that the fastest time to solve these problems is just under four seconds as it took participants on average 3.9 seconds to answer a problem they had previously solved. The main effect of having worked on the problem before also supports the idea that the unsolved problems are more difficult than the unseen problems.

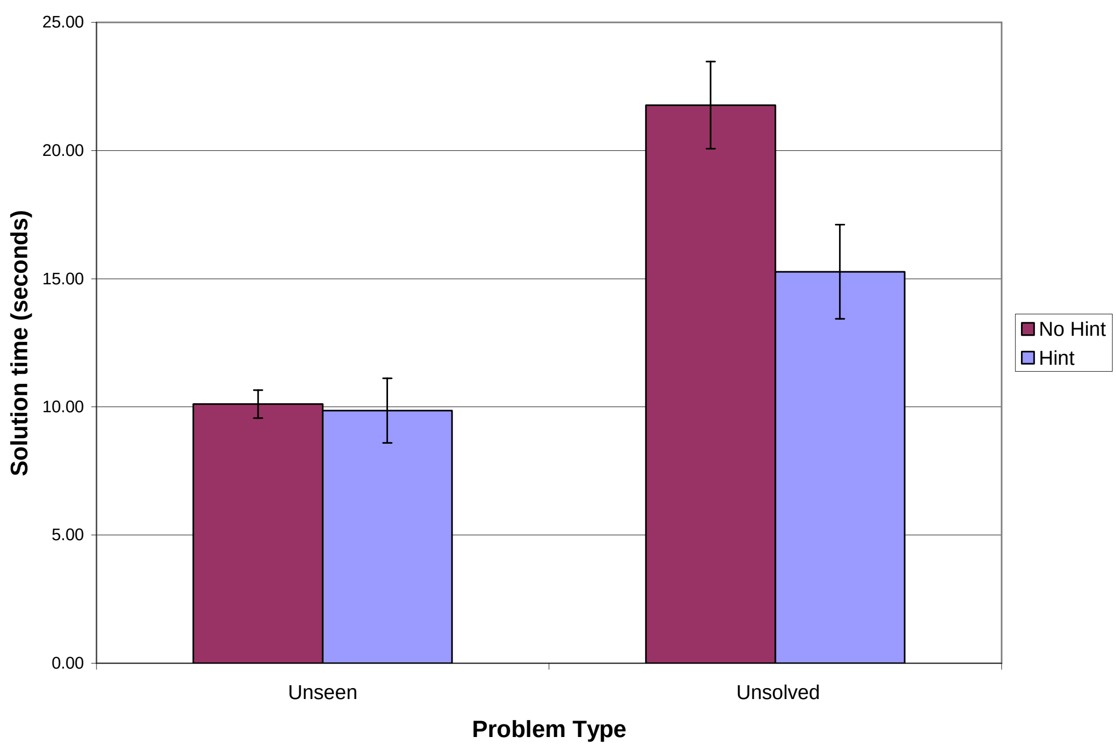


Figure 6. Solution time for solved problems

*Free Recall of RAT Problems*

For the free recall task, each problem recalled by a participant was classified according to whether the participant left the problem *unsolved* every time it was seen, *solved* it every time, or left it unsolved but *improved* by solving it the second time. There were very few cases where a participant solved a problem but then did not solve it the second time. Two participants were excluded from this analysis since they did not improve on any problems, but their exclusion did not change the results for the unsolved and solved conditions. For each class of problems, the proportion of problems recalled from that class is presented in Figure 7. There was a difference in the proportion of problems recalled from each class of problems, $F(2,54) = 5.90$, $p = .005$. Further contrasts show that improved problems differ significantly from both correct problems, $F(1,27) = 8.12$, $p = .008$, and incorrect problems, $F(1,27) = 5.69$, $p = .024$.

While each class of problems was presented an equal number of times, it was the case that participants spent longer working on problems that were left unsolved at some point. The average total time spent working on each problem is presented in Figure 8. While time spent on the problem may be able to explain the difference in recall observed between solved and improved problems, the difference between improved problems and unsolved problems can not be attributed to time as participants spent less time on improved problems. This was confirmed by using a regression analysis to control for the effect of time. The difference in recall between solved and improved problems was no longer significant after time was controlled for. This is to be expected since time was highly correlated with whether the problem was solved or not due to the design of the study.

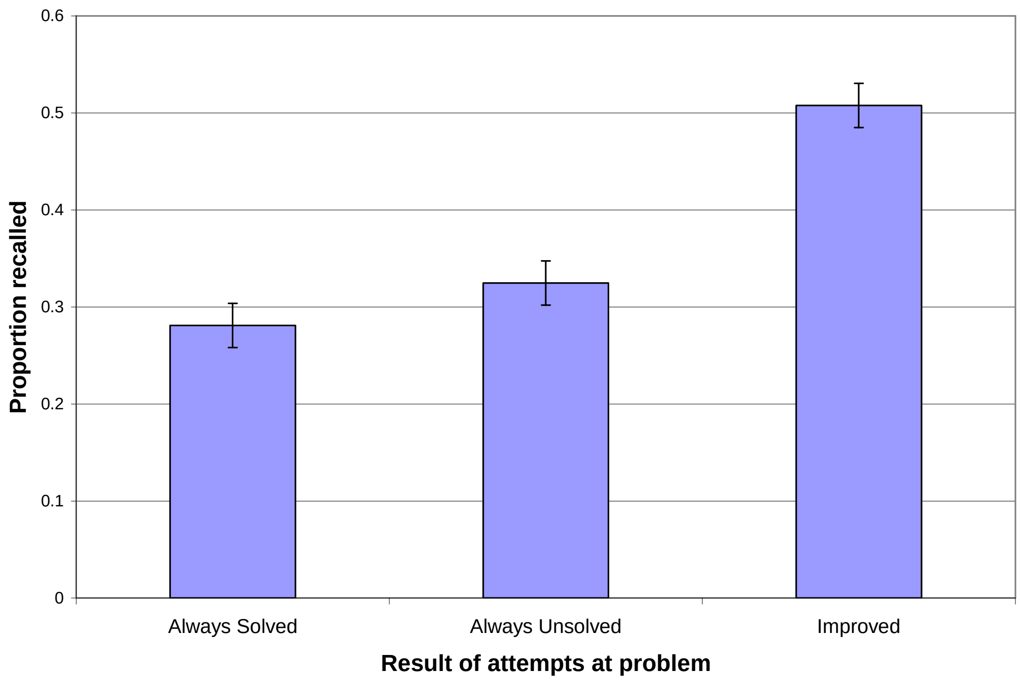


Figure 7. Proportion of each type of problem recalled

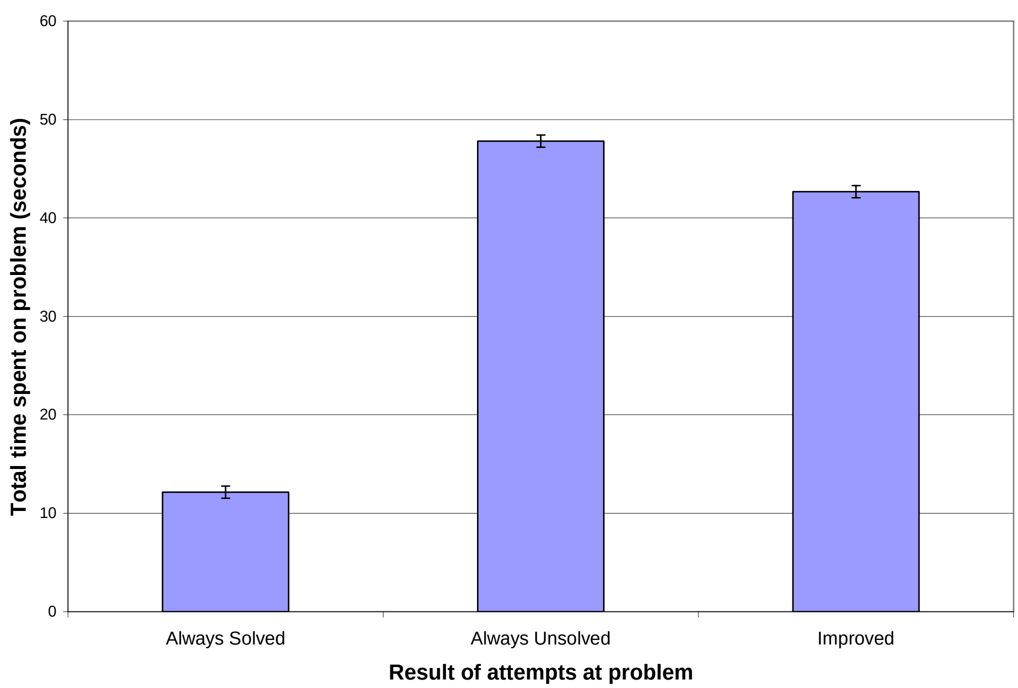


Figure 8. The average total time spent on each class of problem.

There are a few possible reasons for the higher rate of recall for improved problems. One is that these problems represent highly memorable experiences as the problem was solved after initially being unable to solve it. Alternatively, these problems could represent ones where participants had an insight-like experience where the answer just seemed to come to them upon seeing the problem again. Another reason that solved problems could be easier is that they are generally more connected or coherent. Unsolved problems consist or three relatively unrelated words, while solved problems consist of three words, the answer, and the compound words that the answer forms with the problem words. These multiple connected retrieval cues should make solved problems easier to recall.

In order to assess the coherence explanation, the amount of the problem actually recalled was examined. For the recalled problems, participants could have written down one or more of the four words that made up the problem and its solution. The average number of recalled words (excluding the answer) was compared for problems that were always solved, always left unsolved, and improved upon during the second attempt. There were only 19 participants who had recalled at least one problem of each type, and there was no significant difference between these three conditions, $F(2,36) = 1.32$, $p = .28$, with an overall average of 1.77 out of 3 words being recalled. This result means that there was no evidence that solved problems were any more connected than unsolved problems. In other words, accessing one word in a solved problem does not automatically mean that the other words are any more likely to be retrieved than if the problem had been left unsolved.

The recall results for improved problems were analyzed in more detail by looking at whether the improved problems that were recalled differed in terms of whether they were problems where a hint had been provided and whether they differed in terms of solution time.

There were only 12 participants who improved on problems in both the hint and no hint condition, and therefore had the opportunity to recall problems in both conditions. This situation was due mainly to the lack of participants who improved on problems in the no hint condition. For these 12 participants there was no difference in the proportion of improved problems recalled in the hint and no hint conditions with an overall mean 33% of these improved problems being recalled. Therefore, the hint did not affect the likelihood of an improved problem being recalled.

There were 13 participants who recalled some but not all of the problems they improved upon. For these 13 participants it was possible to look at the solution time for improved problems that were recalled versus those that were not recalled. Improved problems that were recalled were solved faster, $M = 11.0$ s, $SD = 6.42$ s, than improved problems that were not recalled, $M = 16.0$ s, $SD = 6.93$ s, $t(12) = 2.51$, $p = .027$. This result could mean that the improved problems that were recalled were problems that were relatively easy to solve or were ones where the answer just popped into mind.

*Awareness of the Manipulation*

Only 8 of the participants reported noticing the relationship between the RAT problems and the lexical decision task, and excluding them does not change the pattern of results. The participants who noticed the effect did show a slightly larger effect of the hint regardless of whether the problem was previously unsolved or unseen. This is what would be expected if they were employing some strategy to recall the words they had seen in the previous lexical decision task. Three different strategies for solving the RAT problems were reported by the participants, and these are explained in detail in the description of the proposed modeling work below. Ten of the participants reported some attempt to continue working on or keep in mind the RAT problems during the experiment. The most common report was that they continued to work on the RAT problem for a few seconds during the feedback and sometimes during the first few seconds of the following problem. The pattern of results does not change with these participants excluded.

### *2.2.3 Discussion*

This study replicates the results from Study 1A and provides evidence that a simple activation based account is not likely to account for the effect of the hint when there is an open goal. If the effect of the hint was due solely to increasing the accessibility of the answer, then there should be a larger effect of the hint for unseen problems than for previously unsolved problems because these problems do not have any activated problem relevant words other than the answer word. The lower solution rates and longer solution times of the previously unsolved problems across both the hint and the no hint condition is an indicator that prior attempts at solving the problems have activated related but incorrect words, or distractors, which interfere with retrieving the answer.

The fact that a majority of the participants did not report any knowledge of the relationship between the two tasks in the study or any strategy to rehearse or remember the unsolved problems makes it unlikely that the results are due to participants trying to strategically recall words that they had seen. Even when those participants who did indicate awareness were removed, the pattern of results did not change.

The solution time metric may be even more sensitive to the effect of the hint and interference as shown by the interaction between hint and problem type. This result shows that when participants solved previously unsolved problems they did so faster if a hint had been presented. However, problems where the hint was presented before the initial problems solving attempt did not show any benefit of the hint. When the success rate and solution time results are considered together they indicate that the hint was more effective in the case where there was an open goal.

While the results from the problem recall task are not a direct replication of the Zeigarnik effect, they are not inconsistent with the idea that open goals or unsolved problems are maintained at a heightened level of accessibility. The results from the problem recall task indicate that problems which were initially left unsolved but later solved were highly memorable. There are a number of possible explanations for these results. It could be that having to leave a problem unsolved causes the problem to be maintained at a higher level of accessibility. The problems that were improved upon could be the ones that were successfully maintained, and so these problems were still at a relatively high level of accessibility at the end of the study.

Another possibility is that solved problems are easier to recall due to the nature of RAT problems. The representation of an unsolved problem is less connected than that of a solved problem. The original problem is just three words with no obvious relationship among them. While two of the words may share some relationship, the three only share one common remote associate which has yet to be discovered. On the other hand, a solved problem has a more connected or coherent representation as the answer links all of the separate words together as well as to the three compound words or phrases that were formed. Solved problems should therefore have an advantage in recall because of the increased number of retrieval cues and the increased coherence of the representation.

The connectedness of the representation of solved problems does not appear to differ from that of unsolved problems given that participants recalled similar proportions of the problem for solved and unsolved problems. Increased connectedness in the representation should mean that it is more likely for the entire problem to be recalled given that a portion of it was recalled. However, it is still the case that solved problems have more cues that could lead to retrieval of the problem.

Given that the improved problems were more likely to be recalled and that the improved problems that were recalled were ones that had been solved faster could indicate that these problems may have been ones where the answer was generated almost immediately after resuming work on the problem. These could have been problems where insight was experienced, and the added surprise or excitement from unexpectedly finding the answer could have made these problems more memorable.

None of these possibilities are in direct conflict with the idea that open goals are maintained at a higher level of accessibility. However, these results do indicate that there is no simple relation between open problem-solving goals and recall of problems. Other work indicates that open goals may be maintained at a heightened level of activation or accessibility (Goschke & Kuhl, 1993; Marsh, Hicks, & Brink, 1998; Patalano & Seifert, 1994), which should mean that the problems associated with these goals should be easier to recall. However there may be other factors that impact recall including the relative ease of accessing the representations of unsolved and solved problems and the subjective experience associated with the problem such as the experience of insight.

It should also be noted that any maintained level of activation did not translate into an observable priming effect in the lexical decision portion of this study, but other studies have shown such a priming effect at shorter time intervals (Beeman & Bowden, 2000; Bowden & Beeman, 1998; Bowden & Jung-Beeman, 2003a; Kihlstrom et al., 1996; Shames, 1994). Overall, the results from Studies 1A and 1B indicate that there is a reliable impact on unsolved problems from information presented in a second task. However, it is not clear how this information later becomes incorporated into problem solving as the brief increase in activation associated with seeing the hint does not fully account for the results. Studies 2 and 3 were designed to explore constraints related to how this mechanism operates.

# Chapter 3

# Number of Open Goals

## 3.1 Study 2

This study seeks to examine two aspects of the mechanism responsible for the influence of open goals on the acquisition of problem relevant information. The first is the role of the number of open goals that are active during the presentation of relevant information, and the second is the type of task in which the relevant information is presented.

In Study 1A a moderate correlation was found between accuracy on the first set of RAT problems and improvement on problems for which a hint was presented. No such correlation was found between initial accuracy and improvement on problems for which no hint was presented. One interpretation of this is that participants who had fewer open problem-solving goals were more likely to notice relevant information. This would indicate that there may be some sort of resource limitation or interference that limits the number of open goals that can direct information acquisition and that having fewer of these persisting goals may increase the chance of noticing relevant information.

This study investigates this issue directly by varying the number of problem goals that are open when relevant information is presented. This is done by giving participants a series of RAT problems, and beginning the lexical decision task once a specified number of these problems have been left unsolved. If there is some limitation to the number of open goals that can be successfully maintained, then the proportion of problems that are improved upon should decrease as the number of open problem goals increases. Another related possibility is that the hints become less effective because attempting to maintain a large number of open goals might actually decrease the chances of any of the hints being recognized. For instance, a person maintaining two of these goals may recognize both answers, but a person with eight goals may not even recognize two of the answers when they are presented.

The results of a study in the literature similar to Study 1A failed to find any improvement unless participants were informed of the relationship between the two tasks (Dodds et al., 2002). One possible reason for this failure is that their task emphasized orthographic rather than semantic properties of the words that were presented. In order to investigate this explanation more thoroughly this study also manipulates the level of processing for the second task. Results from the transfer literature have shown that participants are more likely to recall the source information during the target problem if the source information is also introduced in a problem-solving context (Lockhart et al., 1988). In this study the words in the second task will be presented either in a problem-solving context or not. The basic design is similar to that of Study 1A except for the varied number of open goals and the fact that the lexical decision task was replaced by one of two tasks which were manipulated between participants. One task was a lexical decision task, and the second was to answer a set of simple analogies of the A:B::C:D type.

The hypothesized results were that information from the second task would be more likely to be incorporated into the relevant problems if there were fewer open goals. It was also predicted that the analogy task would lead to a larger increase in improvement than the lexical decision task. This pattern of results would indicate that the mechanism is susceptible to interference from having to maintain multiple incomplete goals, and that it is sensitive to how the hint is processed.

In addition, two measures of working memory capacity were administered to participants in order to look for a relationship between these measures and individual differences in performance especially in the conditions where there were more open goals. Since it does not seem that participants are intentionally trying to keep these problems in mind, no relationship to improvement on unsolved problems is expected. However there is reason to believe that working memory differences may be related to performance on RAT problems in general as discussed in the modeling work in Chapter 5.

### *3.1.1 Method*

*Participants.* There were 120 participants in this study who were all undergraduates at Carnegie Mellon University who completed the study as part of a course requirement. All of the participants were native English speakers.

*Design and Procedure.* The design and procedure were the same as in Study 1A with two exceptions. The number of unsolved problems for each participant was controlled by switching to the second task after the participant had left 2, 4, or 8 problems unsolved depending on the condition. This factor was a between subjects factor with three levels. Since participants in the conditions where there are more unsolved problems would have been working on the RAT task longer than those in other conditions a filler association task was used where participants were asked to generate as many associates to a word as they could. This filler task was chosen since it involved similar processes as those used in generating an answer to a RAT problem, and it was unlikely to produce any kind of open goal. Each trial of the filler task lasted 20 seconds during which participants were generating associates to a single word. For each participant, the words used in the filler task were randomly selected from RAT problems that were not seen by that participant.

In order to control for both the average time between the initial presentation of a problem and the presentation of the hint as well as for the time between the hint and the second presentation of the problem, the filler task was interleaved with the RAT problems. Based on the data from the previous two studies, it was expected that on average, half of the RAT problems presented would be solved. So for every two problems presented, it would be expected that one would be left unsolved. Furthermore, the average solution time for solved problems during the initial presentation of the problem could be expected to be around 10 seconds. Using these estimates, a pair of one unsolved and one solved RAT problem could be expected to take 40 seconds to complete while two filler tasks would also take exactly 40 seconds to complete. Taking the 8 open goals condition as the starting point, the presentation of the filler task for the other two conditions was designed such that the other two conditions can be seen as selecting certain points from the 8 open goals condition in which to present RAT problems.

Figure 9 illustrates this design. For the first attempt at the RAT problems, each RAT section in Figure 9 corresponds to a pair of one unsolved and one solved problem, and each Filler section corresponds to two filler tasks. For the second attempt at the RAT problems, only the previously unsolved problems were presented, and the expected time to solve these problems was 15-20 seconds. In this second presentation of problems, the same design in Figure 9 was followed except that now each RAT and Filler section corresponds to one previously unsolved RAT problem and one filler task respectively. In addition, the final filler task for the 2 open goals condition was never presented since all of the unsolved RAT problems had been presented at that point.

The presentation of the problems was balanced so that the same number of problems was presented early and late in the mixed block of RAT problems and filler tasks. The average time between the initial presentation of the problem and the hint was therefore expected to be roughly equal across the different conditions with varying numbers of open goals. The time between the hint and the second presentation of the problem was controlled for in the same manner.

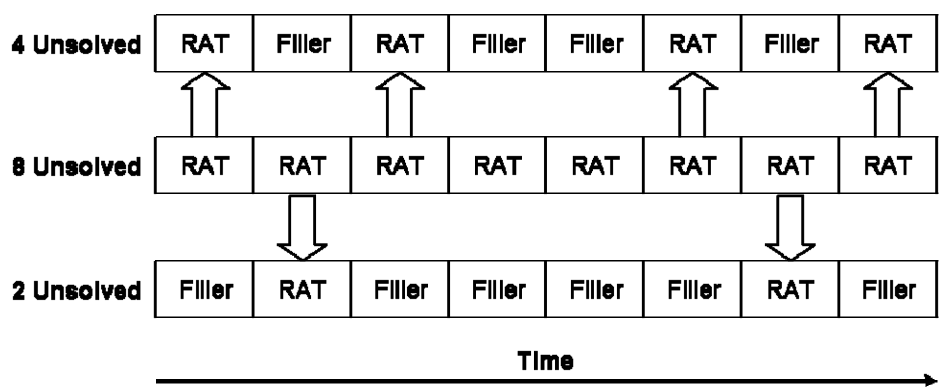


Figure 9. Interleaving of filler task and RAT problems for each condition.

The second factor was the type of task in which the implicit hint was presented. One of the tasks was a lexical decision task, and the other task was a simple analogy task of the form A:B::C:D where the participants chose between two possible answers for D. This analogy task should have caused the participant to process the semantics of the word. In addition the analogy task was in the form of a problem which has been shown to lead to better transfer than a declarative presentation of the same information (Lockhart et al., 1988).

There were two practice RAT problems and two practice word association trials. In addition, there were either three practice analogies or twenty practice lexical decision trials depending on the condition the participant was randomly assigned to. While there were a different number of practice trials for the two conditions, the time spent on the practice trials was roughly equivalent. The RAT problems were presented along with the filler task until the desired number of problems had been left unsolved as described above. The RAT problems that a participant saw were randomly selected from a pool of 29 RAT problems.

In the condition where participants saw the hint in the lexical decision task, there were an equal number of words and nonwords. The initial 10 trials were considered practice trials and no hint was presented during these trials. For the remainder of the task there were 10 filler words and 10 nonwords as well as the hint words and a number of nonwords equivalent to the number of hints presented. The filler words and nonwords were randomly chosen from a larger pool of words and nonwords for each participant.

In the analogy condition participants saw the analogy in the middle of the screen. The missing word in the analogy was always the final word, and a blank appeared in its place. Two possible completion words were presented at the lower left and lower right of the screen. Participants had to press one of two designated keys to indicate which of the two words completed the analogy. The hint was always the correct answer to the analogy. The first five analogies presented were considered practice trials. After these trials, there were 10 filler analogies selected from a pool of analogies as well as one analogy for each hint that was to be presented.

After the intervening task, participants returned to the RAT problems that remained unsolved which were mixed with the word association task as described previously and depicted in Figure 9. After the RAT task had been completed, two measures of working memory were presented. The order of the two working memory tasks was counterbalanced across participants.

The first was a simple digit span task. On each trial of the task, digits were presented visually one at a time for one second each. After the digits were presented, participants were asked to immediately recall the presented digits in order. The number of digits presented on each trial varied from 4 to 9 with three occurrences of each string length. These 18 trials were randomly presented so that participants could not anticipate the number of digits that would be shown on a particular trial. There was one practice trial to give participants a chance to become familiar with the task.

The second working memory task was the modified digit span (MODS) task (Lovett, Reder, & Lebiere, 1999). This task is similar to the digit span task except that the task is designed to eliminate the ability of participants to rehearse the digits that they will be asked to recall. A series of letters and digits are presented to participants at a rate of two per second. The participants are instructed to read aloud each digit and letter as they appear while also remembering the digits in the order they were presented. Each digit is presented at the end of a sequence of letters which all appear at one point on the screen. The next sequence of letters and a digit is presented at the same point but just to the right of the previous sequence. At the end of a trial, participants are asked to recall the digit that appeared at a particular location on the screen as shown in Figure 10. The digits had to be recalled in order from left to right, and once a digit was entered participants were not allowed to go back and change it. The number of digits to be recalled varied from 3 to 6. There were six trials for each digit sequence length. The trials were all randomly ordered so that participants could not anticipate the number of digits that would be shown on a particular trial. There were two practice trials to give participants a chance to become familiar with the task.

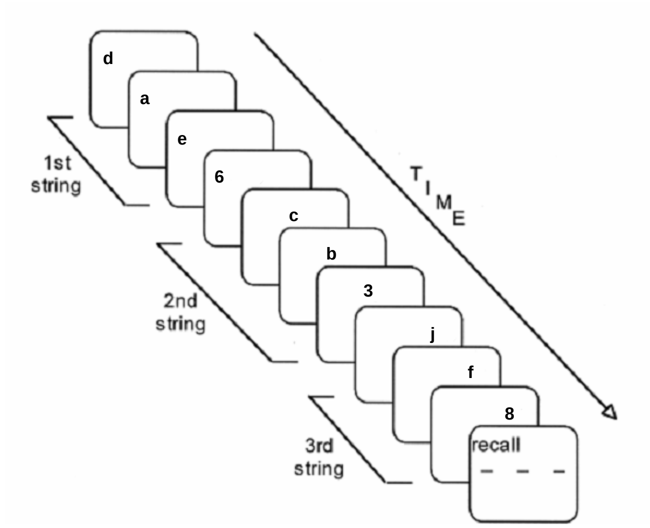


Figure 10. Structure of the MODS task (adapted from Daily, Lovett, & Reder, 2001)

*3.1.2 Results*

Accuracy was above 95% in every condition for both the lexical decision and analogy tasks. Response times for the analogy task were of course longer than for the lexical decision task. The response times for correct trials for both the analogy and lexical decision task were analyzed in 3 x 2 ANOVAs with the number of open goals being a between subjects factor and whether the trial was a hint or filler trial being a within subjects factor. There was no difference in response time between the hint and filler conditions in the analogy task, $F(1, 57) = .045$, and the overall mean response time was 4.43 s ($SD = 1.36$). There was also no main effect of the number of open goals and no interaction. For the lexical decision task, there was also no main effect of the number of open goals and no interaction, but there was a significant difference between the response times for hint and filler trials, $F(1, 57) = 9.08$, $p = .004$. The response times for the lexical decision task are presented in Figure 11. This priming effect for unsolved problems is a replication of the same priming effect that was found in Study 1B. Answers to RAT problems that were not solved were processed faster than neutral words.

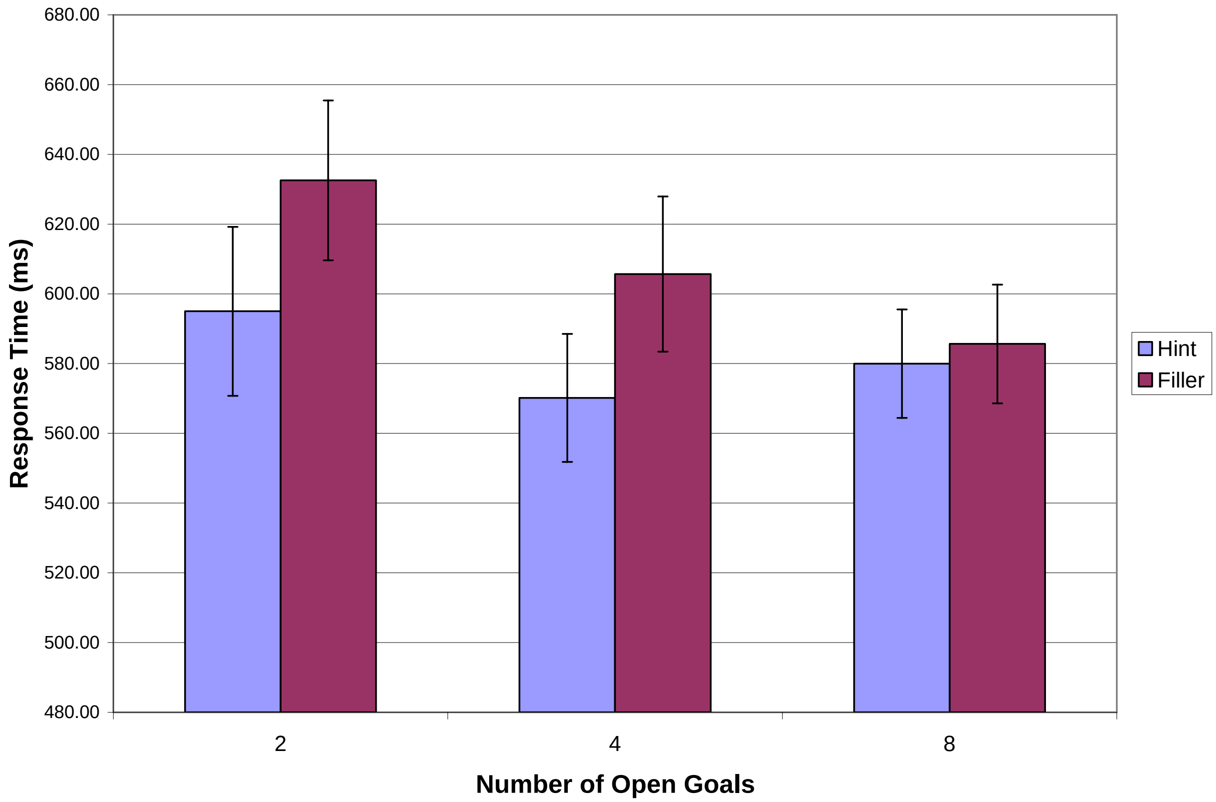


Figure 11. Response times for lexical decision task

The proportion of problems improved upon for each condition is presented in Figure 12. The data were analyzed using a 3 x 2 ANOVA . There were no significant main effects or interactions. Since this study differs from the previous studies by using a between subjects design, the data were also analyzed with the proportion of RAT problems solved in the first block of RAT problems as a covariate to control for individual differences in RAT problem-solving ability. There were again no significant main effects and no interactions.

This analysis was repeated three times with each of the three working memory scores used as a covariate in the analysis. None of these analyses differed from the analysis reported above. Details on the calculation of the working memory scores and their relationship to RAT problem solving can be found in the working memory section below.

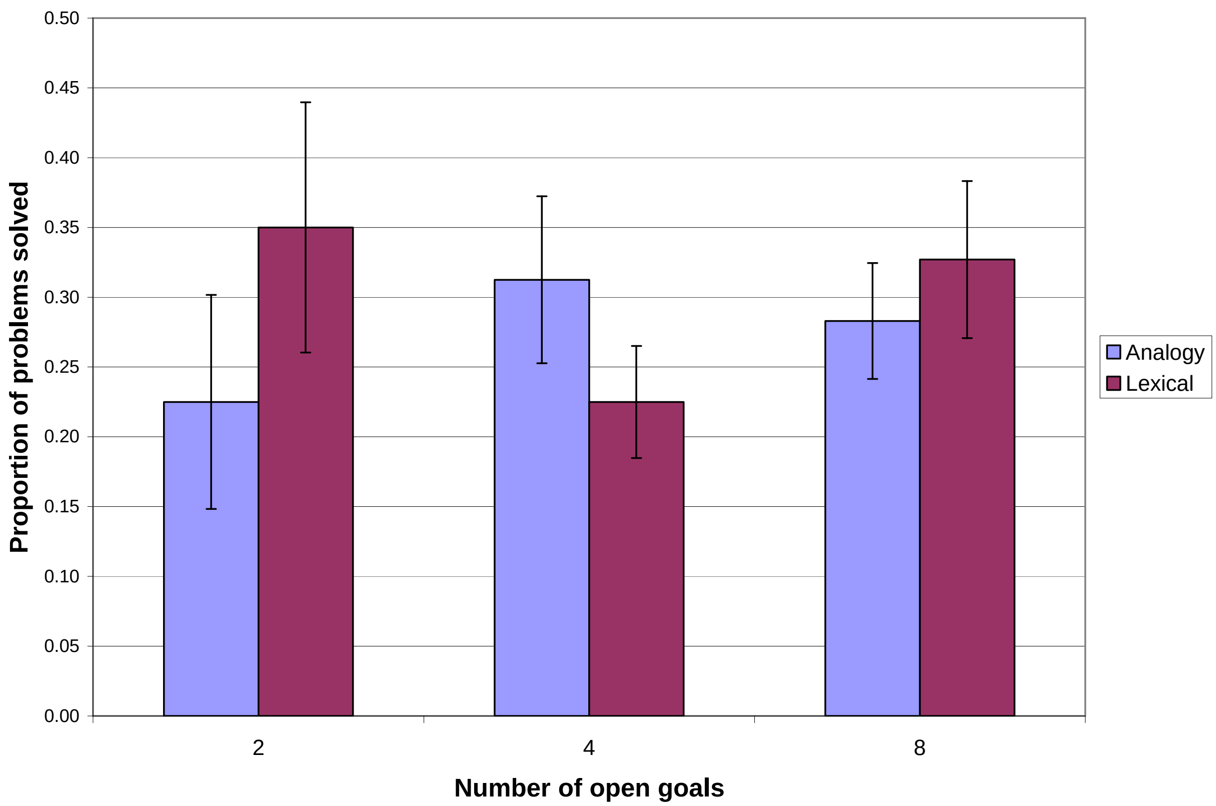


Figure 12. Effectiveness of the hint

*Filler Task*

The word association task was included to control for the amount of time between the end of an unsolved problem and presentation of the corresponding hint as well as the time between the hint and the second attempt at the problem. This aspect of the study design was based on solution rate and time data from previous studies. Figure 13 shows the average amount of time from the presentation of a problem to the hint and the time from the hint to the second presentation of the problem. Overall, the design worked fairly well as the maximum time difference is about 34 s. These times were analyzed with a 3 x 2 ANOVA with number of open goals and task type as factors. For the time between the initial presentation of the problem and the hint there was a main effect of task type, $F(1, 114) = 7.44$, $p = .007$. There was no effect of number of open goals and no interaction. There was a 10.8 s difference between the lexical decision and analogy task conditions. For the time between the hint and the second presentation of the problem, there were main effects of task type, $F(1, 114) = 8.64$, $p = .004$, and number of open goals, $F(2, 114) = 25.95$, $p < .001$. There was no interaction. There was a 10.5 s difference between the task types in the opposite direction of the previous 10.8 s time difference. The main problem is that there was an increasing amount of time between the hint and the second presentation of the problem as the number of open goals increased. This means that there could be a confound as time and the number of open goals were not independent, but this is only a problem if the confound caused the lack of a result in the proportion of problems improved upon as shown in Figure 12.

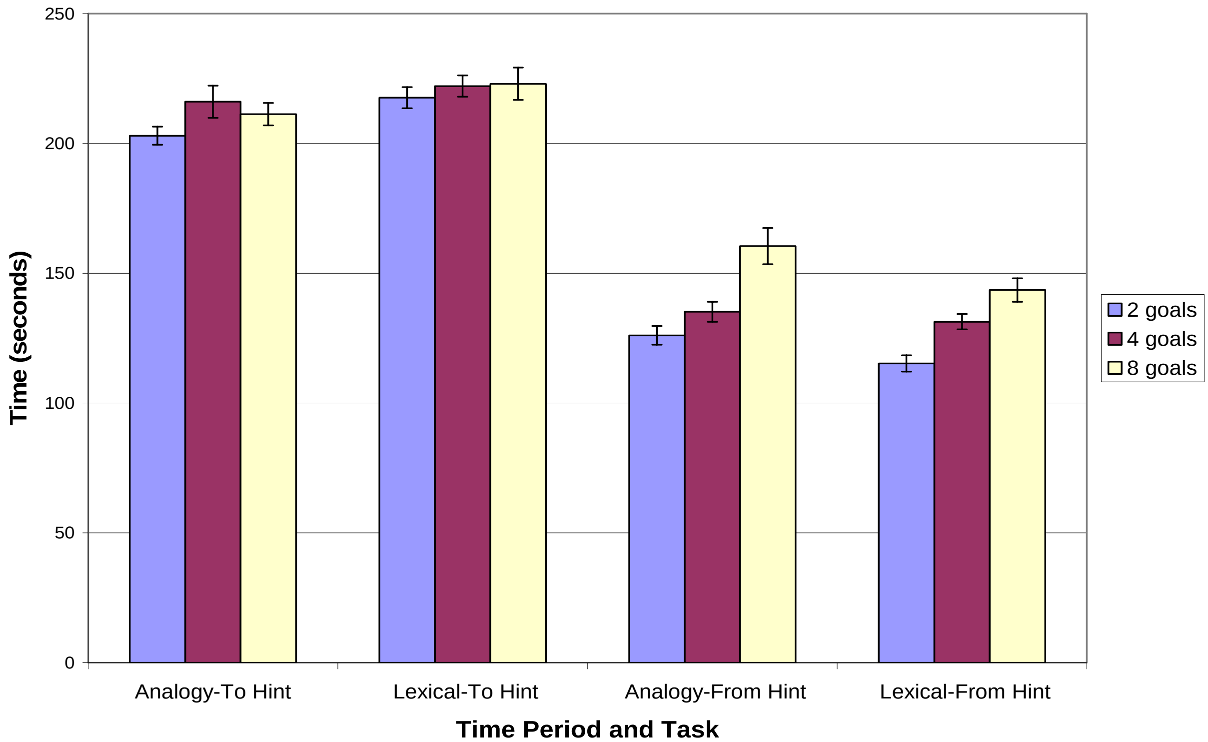


Figure 13. Time between first and second presentation of problems and hint

This confound is not likely to have been a major problem as the there was only a maximum of about a 30 s difference between the conditions, and this was not a particularly long time given that the amount of time spent on a single unsolved problem is also 30 s. There was a statistically significant correlation of -.378 between the proportion of problems improved upon and the amount of time between the hint and the second presentation of the problem. This correlation can be explained by the fact that the study design was based upon an expectation of 30-50% of previously unsolved problems being improved upon, and the actually average proportion of problems improved upon was consistently near the lower end of this range. While the proportion of improved problems remained the same across conditions, the total amount of time spent on the second set of RAT problems increased along with the increasing number of open goals.

The alternative interpretation of this correlation is that the increasing amount of time between the hint and the problem led to a decrease in the effectiveness of the hint. This would mean that there was actually an increasing effect of the hint as the number of open goals increased, and the increase in the amount of time between the hint and the problem was responsible for causing the amount of improvement to appear constant across the different open goal conditions. However, no relationship was ever found between the amount of time since the hint and the likelihood of improving on a problem in either Study 1A or 1B.

There is one plausible reason that the effectiveness of the hint could increase as the number of open goals increased. Due to the design of the study, the proportion of hints in the lexical decision and analogy tasks did increase as the number of open goals increased. This is because the hints were added to a number of filler trials that was constant across conditions. If participants had been trying to recall the words in the intervening task when they were working on the second presentation of RAT problems, then it would be expected that an increase in the proportion of hints to non-hints in the second task would lead to an increase in the RAT problems improved upon. This situation is only likely to occur if participants had caught on to the relationship between the two tasks in the study.

At the end of the study, participants were asked if they noticed the relationship between the RAT problems and the lexical decision or analogy task. Only 16 of the 120 participants were able to report the relationship. There was no relationship between the participants who noticed the relationship and condition they were assigned to. When these participants were removed from the data, the correlation between improvement and the time between the hint and the second presentation of the problem remains relatively the same, r = -.332, p = .001. These findings support the interpretation that the correlation was caused by a constant rate of improvement across conditions with the filler task duration not being long enough to compensate for the time spent attempting to solve RAT problems during the second presentation of the RAT problems.

The number of associates generated in the filler word association task was examined to insure that participants were actively working on the task. Overall, participants in conditions with the filler task generated 4.67 associate words per association trial (SD = 1.46). This means that participants were generating a new association about once every four seconds.

*Working Memory*

The working memory measures were scored in two ways. The first, which is referred to as all or nothing credit, basically counts the number of times a participant recalled the entire recall trial correctly. This measure is just the total number of completely correct trials from the working memory task with each task having an equal number of trials at each digit sequence length. The second method is referred to as partial credit and calculates the proportion of digits that were recalled in the correct serial position for each trial. These proportions are then summed to create the working memory score for each participant. Due to a technical error in the data collection, the digit span task was only able to be scored using the all or nothing method.

Due to technical difficulties, the data from the MODS task were not recorded for one participant. The working memory measures were first compared to the proportion of RAT problems solved on the first presentation of the problems. None of the working memory measures correlated significantly with this measure with the correlations ranging from .019 to

.069. In order to determine if there was any relationship between working memory and hint effectiveness, the working memory measures were correlated with the proportion of problems that were solved in the second attempt that had not been solved on the first attempt. Again none of the measures correlated significantly with the correlation ranging from -.106 to .056.

Since those participants in the 8 open goals condition would have attempted the most problems in the first set of RAT problems their performance on these problems was correlated with the working memory measures as a separate group. None of these correlations are significant with the correlations ranging from -.066 to .009. In addition, if there were a relationship between working memory and a limit on the number of open goals for which the hint was effective, those participants in the 8 open goals should have shown the strongest relationship. To assess this relationship, the working memory measures were correlated with the proportion of problems solved after the hint had been presented. In this case, there was a significant correlation of -.316, $p = .047$, with the digit span measure with the correlations ranging from -.316 to -.211. If one outlier participant is removed, the correlation is reduced to -.269, $p = .098$, and is only marginally significant. These results indicate working memory related weakly or not at all to RAT problem solving and the effectiveness of the hint.

*Feelings of Warmth*

Feeling of warmth ratings for problems that were initially left unsolved were analyzed in order to compare problems which participants eventually improved on to those which remained unsolved. There was no difference in feeling of warmth for improved and unsolved problems with the overall mean rating being 1.65 (SD = 0.66). These results indicate that participants were unable to indicate which problems they were closest to solving, and they generally rated feeling of warmth to be low for unsolved RAT problems.

### *3.1.3 Discussion*

The results of this study support the idea that open goals are not being actively maintained with some kind of limited resource. There seems to be a fixed chance that the information in the hint will be used in subsequent problem solving. This result is compatible with a modified form of the memory sensitization hypothesis (Yaniv & Meyer, 1987; Yaniv et al., 1995). The original hypothesis held that trying to retrieve a particular word that led to a tip of the tongue state or high feeling of knowing would sensitize memory so that future encounters with the word would increase the chance of later recalling the word successfully above and beyond the usual increase in recall probability associated with just seeing the word without the failed retrieval attempt.

The current study uses simple insight-like problems to show that a similar type of phenomenon applies even when the sought after word is not a simple memory retrieval, and one where participants are unable to generate a feeling of warmth that is predictive of their chances of solving the problem. This result contrasts with the results of the studies in the literature used to support the memory sensitization hypothesis where a measure of feeling of knowing was predictive of hint effectiveness (Yaniv & Meyer, 1987; Yaniv et al., 1995). In these prior studies examining word recall and in the current studies with RAT problems, a word is trying to be recalled in order to satisfy the constraints of the task. However, the RAT problems require search for a solution as opposed to the retrieval of a word from a definition. In both cases, elements in memory are somehow modified to make it more likely that relevant information encountered in the meantime is used when that problem or retrieval task is encountered in the future. The lack of any relationship of improvement to feelings of warmth is consistent with findings that feelings of warmth on insight problems are not predictive of solution (Metcalfe, 1986a, 1986b).

The lack of a significant relationship between working memory and hint effectiveness argues against an active maintenance of open goals in working memory. When combined with the fact that few of the participants in Study 1B indicated trying to remember problems or trying to persist in their efforts to solve problems after the problem was gone, the lack of a relationship to working memory further rules out some kind of conscious effort to rehearse or maintain unsolved problems. These results are consistent with open goals being stored in memory without active maintenance, but nonetheless information relevant to the open goals is more likely to be used when it is encountered.

Working memory was not related to performance on the initial set of RAT problems. Others have found low to moderate correlations between RAT performance and working memory measures (Kane et al., 2004). However, these correlations were found with versions of RAT problems that were not restricted to compound words and phrases (Bowden & Jung-Beeman, 2003b). Given that the working memory measure used here has been related to a parameter in the ACT-R architecture (Daily et al., 2001; Lovett, Daily, & Reder, 2001; Lovett et al., 1999), it will be discussed in relation to the results of the model below.

# Chapter 4

# Protocol Study

## 4.1 Study 3

Studies 1A and 1B demonstrate that presenting hints in a second task was most effective in aiding problem solving if the problem had been attempted and left unsolved. Study 2 shows that this phenomenon is not related to the active maintenance of open goals with some highly limited resource. The purpose of Study 3 is to examine the effectiveness of hints in relation to the representation of the unsolved RAT problem at the time problem solving is suspended.

Verbal protocols were recorded while participants solved these problems. These protocols were used to assess the strategies that participants were using to solve these problems as well as the number and types of words and concepts they explored. One can imagine that as problem solving progresses participants will explore an increasing number of possible words. They can also be expected to explore different possible meanings for the words that make up the RAT problem. This exploration is not likely to be random and will be guided by the meanings and concepts associated with one or more of the words which make up the RAT problems. For example, with the word “tail” there are a number of meanings that could be explored including animal tails or the more general meaning of something from behind as in the word “tailwind”.

One possibility is that hints are more likely to influence problem solving when more of the related concepts have been explored. Another possibility is that the hint could be more effective if participants have thought about the meaning or context that is most congruent with the meaning of the sought after compound words or phrases. It could also be that reaching an impasse in problem solving makes it more likely that the hint will influence work on the problem. It has been suggested that people are more likely to remember unsolved problems when an impasse has been reached (Patalano & Seifert, 1994). The likelihood of an impasse being reached can be manipulated by varying the amount of time participants are given to work on the RAT problems.

An impasse in problem solving is a situation where the problem solver does not know how to proceed. Impasses may be resolved in many different ways, but the key characteristic of an impasse is a period of time where the problem solver does not make progress on the problem even though the problem solver is actively trying to solve the problem. The operational definition of an impasse that will be used in this work is the point at which participants are unable to generate any new candidate words for more than 10 seconds or where they indicate verbally their inability to generate new possibilities. The likelihood of reaching an impasse increases with the amount of time the participant has spent on the problem.

In addition to being more likely to reach an impasse with increasing time it is also the case that spending more time on the problem allows participants to develop a better representation of the problem by exploring more of the words and concepts related to the problem. In order to manipulate the development of the representation for a problem varying time limits were imposed. The imposed time limits provide some control over the development of the representation of the RAT problem including whether an impasse was reached. The verbal protocols were equally important as they provided a way to classify whether an impasse in problem solving occurred regardless of the amount of time spent on the problem. In other words, impasses will be most likely to be reached at the longest time limit, but the verbal protocols will insure that an impasse has been reached on a problem by problem basis so that the results could be analyzed both in terms of the concepts explored and whether an impasse has been reached.

In addition, a word recall task was conducted after the rest of the experiment was completed. Verbal protocols were recorded while participants were asked to recall a low frequency word based on its definition. The purpose of this definition task was to provide a set of protocols to compare to the protocols taken from the RAT problems in order to demonstrate how the search process in RAT problem solving were distinct from a basic memory retrieval task.

### *4.1.1 Method*

*Participants.* There were 30 participants in this study who were all undergraduates at Carnegie Mellon University who completed the study as part of a course requirement. All of the participants were native English speakers.

*Design and Procedure.* The design and procedure were similar to Study 1A with a few additions related to verbal protocols and impasses. First, participants were instructed on how to give concurrent verbal protocols using the procedure and practice tasks found in the appendix of Ericsson and Simon (1993). Participants were instructed to think aloud as they worked on the RAT problems. They were also told that they did not need to try to verbalize during the lexical decision task. The initial set of RAT problems contained 30 problems which were equally divided into three conditions where the time limit for working on the problem was either 15, 30, or 60 seconds.

After participants were given instructions on how to think aloud they were presented with instructions for the RAT and lexical decision tasks. They were told that they would alternate between these two tasks throughout the study. They were then given three practice RAT problems and 20 practice lexical decision trials before the study began. The participants then began work on 30 RAT problems presented one at a time. The problems were presented with 15, 30, or 60 second time limits. Participants could type in their answer at any time during the problem. If the attempted answer was not correct the answer was cleared, and they could continue working on the problem for the remainder of the time limit. After the RAT problems had been presented, the lexical decision task was presented. The first 10 trials of the lexical decision task were considered practice trials, and none of the implicit hints were presented during these trials. The words consisted of 10 filler words that were unrelated to the RAT problems as well as answer words to half of the unsolved RAT problems at each level of the time limit factor. The number of nonwords appearing in the lexical decision varied depending on how many hint words had to be presented, but in all cases the number of words and nonwords were equal. Presentation order was randomized for each participant.

After the lexical decision task was presented, participants were informed that they would see the same 30 RAT problems they had worked on previously. Each problem was presented with a time limit of 30 seconds regardless of how long the problem had been presented initially. Participants were then given a surprise recall task where they were asked to recall as many of the RAT problems as they could remember. They were instructed to recall as much of each problem as possible, but they were told that it was fine if they could only remember part of the problem.

Participants were then asked three questions. The first question asked if they had noticed any relationship between the lexical decision and RAT problems. The second question asked them to describe how they worked on the RAT problems including any strategies they may have used. The final question asked if they had ever tried to rehearse or keep working on any of the RAT problems after the time for that problem had expired.

### *4.1.2 Results*

*Lexical Decision Task*

One participant who averaged 50% correct during the lexical decision task was excluded from all analyses. The remaining participants averaged 97% correct for the lexical decision task, $SD = .03$. The response times for the correct responses to the words in the lexical decision task were analyzed. In this analysis responses that were faster than 300 ms or slower than 1200 ms were excluded. The hint words, which were answers to previously unsolved problems, were responded to slightly faster ($M = 603$, $SD = 88$) than the filler words which were unrelated to the RAT problems ($M = 621$, $SD = 101$), and this difference was marginally significant, $t(28) = 1.75$, $p = .09$.

*Improvement on RAT Problems*

The proportion of RAT problems answered correctly on the first attempt is presented in Figure 14. Spending longer on the problem did lead to an increasing proportion of problems being answered on the first attempt, $F(2, 56) = 8.86$, $p < .001$. Further contrasts showed that the 60 second time limit differed significantly from both the 15 s, $F(1, 28) = 14.5$, $p = .001$, and 30 s time limits, $F(1, 28) = 10.5$, $p = .003$. The 15 s and 30 s time limits did not differ.

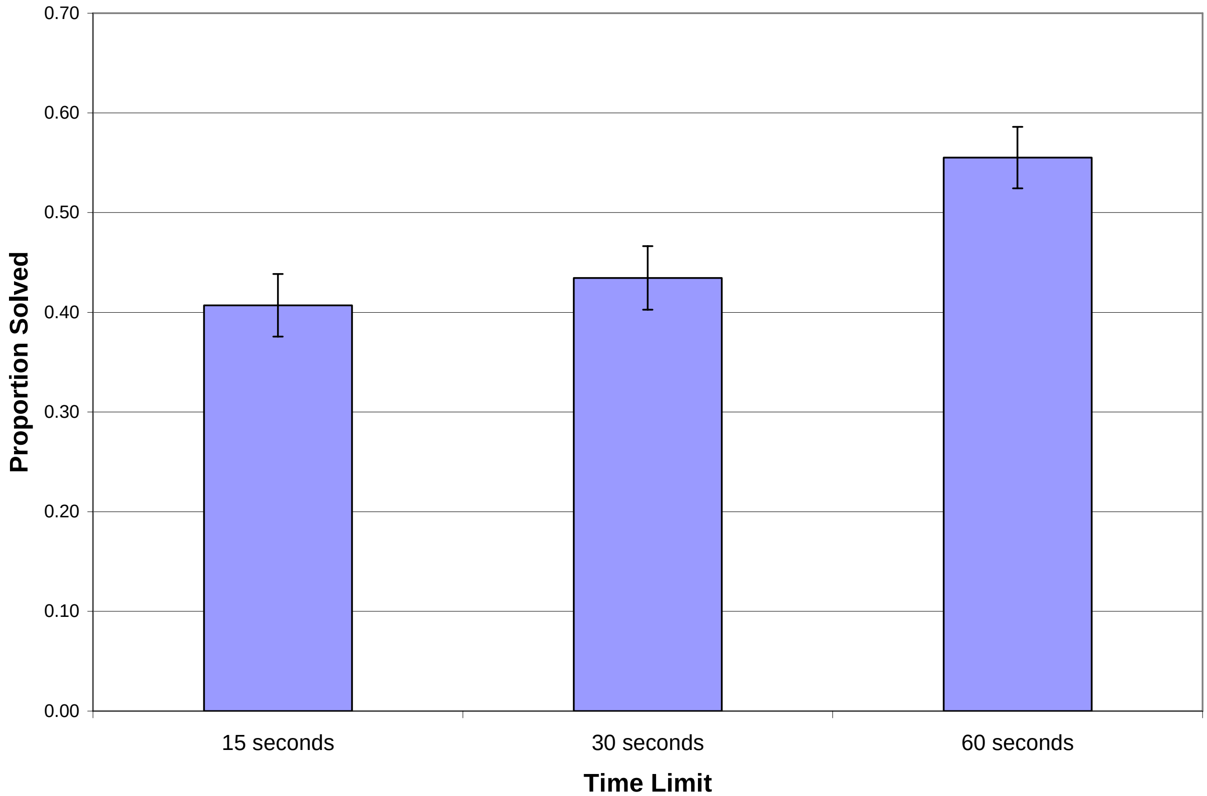


Figure 14. The proportion of RAT problems answered correctly on the first attempt.

The improvement on previously unsolved RAT problems is presented in Figure 15. A 3 x 2 within subjects ANOVA was run with time limit and hint as factors. Two subjects were excluded from this analysis because they were missing data in one or more cells. This happened when there were fewer than two remaining unsolved problems at a particular time limit. There was no main effect of time limit, $F(2, 52) = 1.25$, $p = .30$, or hint, $F(1, 26) = 1.19$, $p = .29$. There was also no interaction, $F(2, 52) = .91$. The lack of significant effects may be due to the limited power of the study. The number of participants was limited on purpose as the collection and analysis of verbal protocols is time consuming.

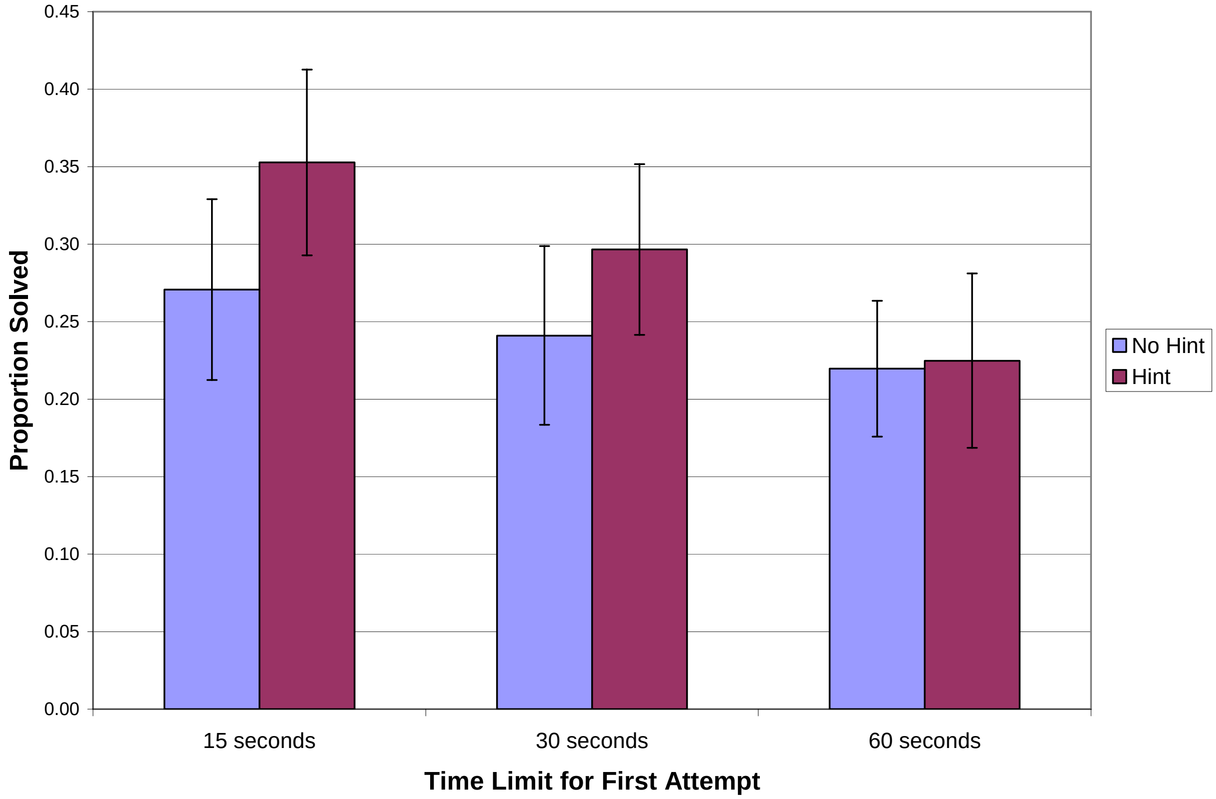


Figure 15. The effect of hints on second solution attempt for unsolved RAT problems

The solution times for previously unsolved RAT problems that were improved upon were also analyzed and are presented in Figure 16. There were too many missing cells to break the solution time data down by time limit so the data were collapsed across the three time limits. It should be remembered that there was always a 30 s time limit for the second presentation of a problem so the solution time measures that are being collapsed all have the same potential range. One participant did not improve on any problem and so was excluded from this analysis since there was consequently no solution time data for that participant. The only remaining missing cells were in the no hint condition so replacing these five cells with the maximum time of 30 s should not affect the analysis since if anything it provides an underestimate of the solution time in the no hint condition. Previously unsolved problems were solved faster in the hint condition, and this difference was marginally significant, $t(27) = 1.93$, $p = .06$.

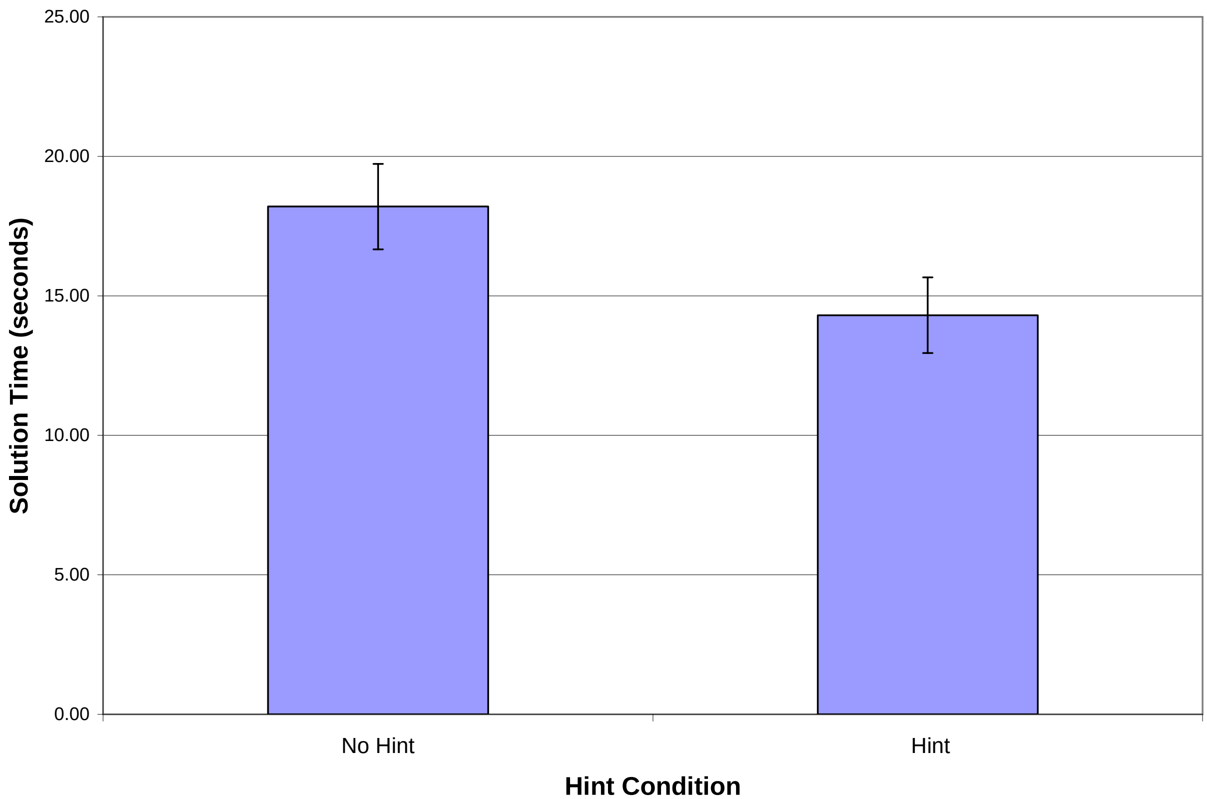


Figure 16. Solution time for previously unsolved problems collapsed across time limit

*Impasses*

The verbal protocols were used to determine when participants reached an impasse in their problem solving. While it is increasingly likely that participants reached an impasse as the time allowed for a problem increased, the protocols provide a method for detecting impasses on a per problem basis. For each problem the times at which a participant introduced a new candidate answer were recorded. A new candidate answer is a word that has been retrieved and verbalized that had not already been verbalized in that problem attempt. An impasse was defined as a period of more than 10 seconds without generating a new candidate answer. The one exception to this definition was at the start of the problem where an impasse was defined as 15 seconds without generating a candidate. The additional 5 seconds was to allow participants to encode the problem and begin work on it. The first attempt at a problem was classified either as an instance of impasse or not. Problems within the impasse category were further coded for whether or not the impasse lasted until the end of the time limit for that problem. It could be the case that overcoming the impasse and continuing to work on the problem may be different from ending the problem at an impasse.

The proportion of problems solved on the second attempt is broken down into impasse conditions in Figure 17. There was no apparent difference in proportion solved for the two different impasse types that were defined. All further analyses will only deal with the case where an impasse occurred regardless of where in the problem solving it occurred. Two participants were excluded from the analysis because they did not have data for all four conditions. A 2 x 2 within-subjects ANOVA was used to analyze this data with the presence of an impasse and hint as the two factors. There was a marginally significant interaction, $F(1, 26) = 3.02$, $p = .09$. The hint was effective only in the case where there was no impasse.

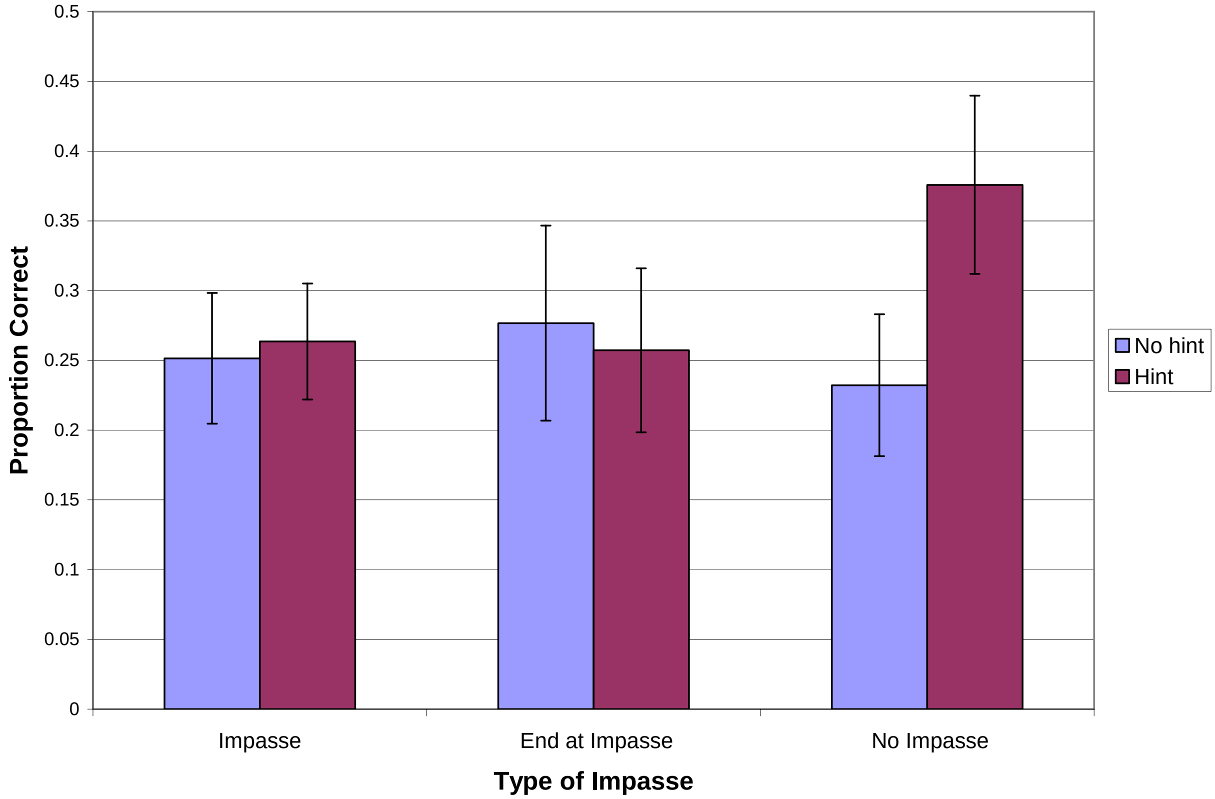


Figure 17. Proportion of problems solved by type of impasse

*Fixation*

Fixation on distractor words was examined between problem attempts. Fixation was defined in terms of the number of times distractor words that were generated and rejected on the first problem-solving attempt were generated in the second problem-solving attempt. An example of fixation can be seen in Figure 18 where the participant repeats some of the same distractors both within and across problem-solving attempts. In this case, the participant is trying to solve the problem “gear”, “hammer”, “hunter” for which the answer is “head”.

Because there were more distractors generated during the problems which were initially presented for a longer period of time, two proportion measures were used. The first measure was the proportion of candidates from the first attempt that were repeated in the second attempt. The second measure was the proportion of candidates in the second attempt that were repeated candidates. Figure 19 presents results using the first measure, and Figure 20 presents results using the second measure.

**First Attempt**

A hammer
A deer hunter
A jack hammer
um
rubber hammer
gear
mountain gear
duck hunter
duck gear
(types duck)
Um
a
There's a jack hammer
There's a rubber hammer
There's

**Second Attempt**

uh
hammer
a jack hammer
a
carpenter's hammer
a
mmm
hunter
hunter

Figure 18. Example protocol showing fixation

Within-subjects ANOVAs were used to analyze both measures with hint and the result of the second problem attempt as factors. There was one participant who never improved on a problem and seven who did not improve in the condition without a hint. These participants were excluded from the analyses. For the proportion of candidates in the first attempt that were repeated in the second, shown in Figure 19, there was a marginally significant interaction, $F(1,21) = 4.19$, $p = .053$. In addition there was a main effect of improving on the problem, $F(1,21) = 20.86$, $p < .001$ but no main effect of hint, $F(1,21) = .663$. For the proportion of candidates in the second attempt that were repeats, shown in Figure 20, there were main effects of hint, $F(1,21) = 5.53$, $p = .028$, and improving, $F(1,21) = 13.86$, $p = .001$. There was not a significant interaction, F(1,21) = 1.10. Both of these measures demonstrate that fixation on distractors is a source of difficulty and that the hint was effective in overcoming fixation in some cases.

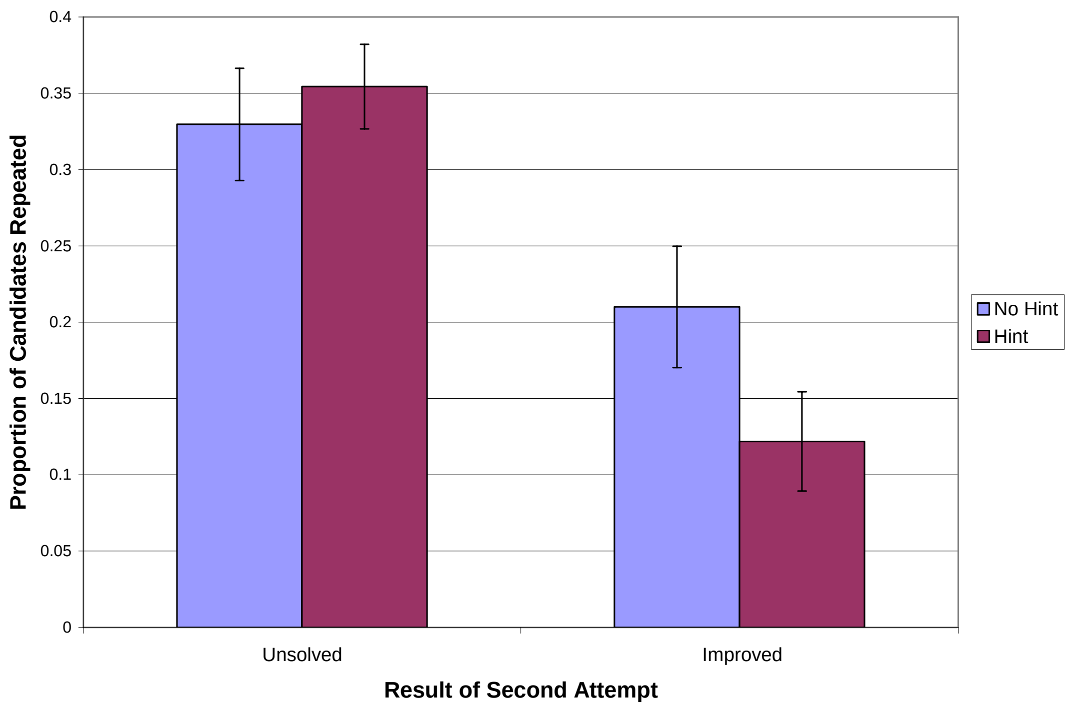


Figure 19. Proportion of candidates that were repeated on the second attempt

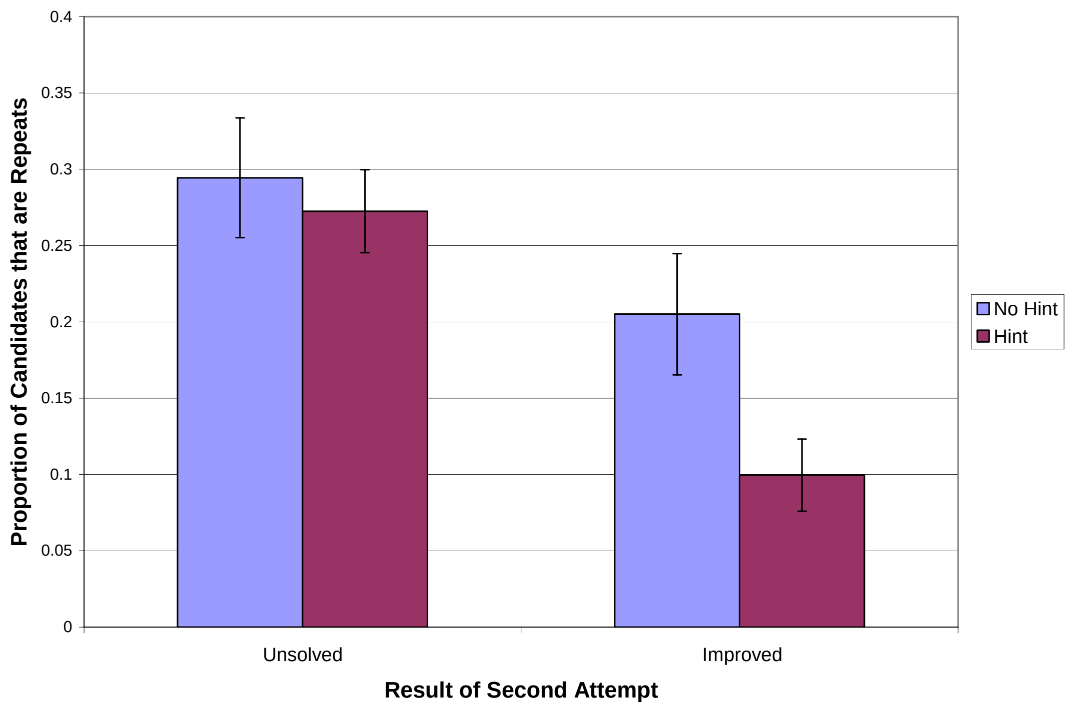


Figure 20. Proportion of candidates generated in second attempt that were repeats

Fixation can also be examined in terms of whether an impasse has been reached. Figures 21 and 22 present the two measures of fixation in terms of impasses and whether a hint was presented. One participant did not encounter the no-hint condition on a problem with an impasse and so was excluded from this analysis. For the proportion of candidates from the first attempt that were repeated there was a main effect of impasse, $F(1,26) = 12.73$, $p = .001$, but no significant effect of hint, $F(1,26) = 1.99$. There was no interaction, $F(1,26) < 1$. For the proportion of candidates in the second problem that were repeats, there was a significant effect of hint, $F(1,26) = 5.71$, $p = .024$, but no effect of impasse, $F(1,26) < 1$. There was also no interaction, $F(1,26) < 1$.

Fewer of the candidates from the first problem-solving attempt were repeated in the case when an impasse was reached. This could indicate that after reaching an impasse participants are taking a qualitatively different approach to the problem which could also relate to the fact that the hint was less effective when an impasse was reached. The results from Figure 22 show that the hint reduced the proportion of candidate words on the second attempt that were repeats. This is partially due to the fact that participants improved more with a hint and so probably generated the solution early in the problem. However, the improvement was only found in the non-impasse case, and the hint appears to have been somewhat effective in reducing fixation in the impasse case as well.

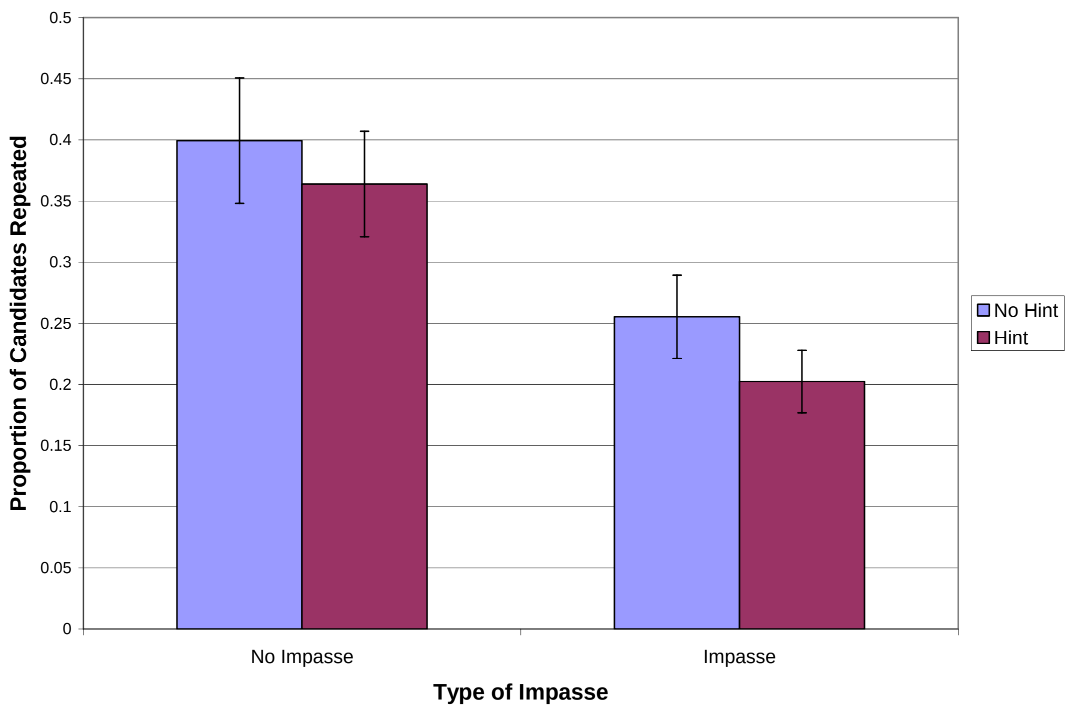


Figure 21. Proportion of candidates from that were repeated (by impasse condition)

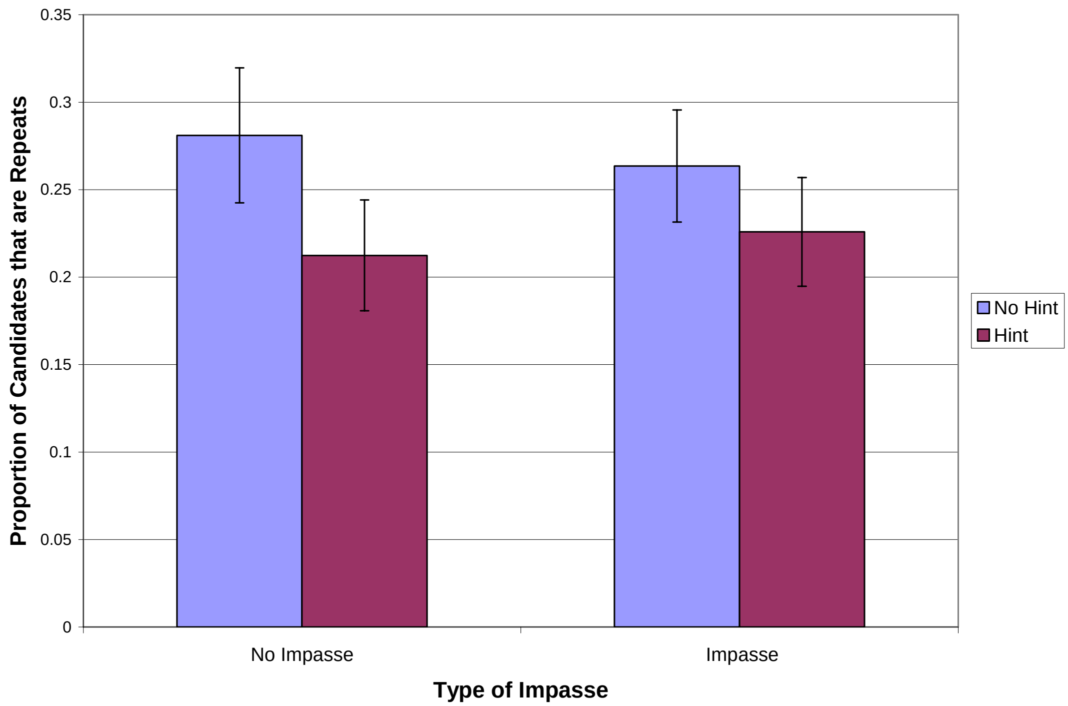


Figure 22. Proportion of candidates that were repeats (by impasse condition)

*Impasses and Fixation*

The operational definition of an impasse used in this study is a period of time, 10 s, without producing a new candidate word. Participants did not usually stay silent during this period of time as they continued to work on the problem. Some of this time will be spent repeating previously tried candidates which should lead to an increase in fixation. This increase in fixation during an impasse may be one reason why the hint was only effective for problems in which there was no impasse.

One measure related to this fixation explanation that was obtained from the protocols is the number of unique candidates generated during an attempt at the problem. As more time is spent generating new candidates, there is less time that is spent regenerating old candidates. The number of candidates generated on the first unsuccessful problem attempt was used as a continuous variable along with the time limit and hint factors in a logistic regression where the outcome variable was whether the second problem attempt was successful. In other words, is the number of unique candidates a good predictor of improvement along with hint and time limit?

The resulting fit from the logistic regression is shown in Figure 23. There was a marginally significant main effect of number of unique candidates, $\beta = -.186$, $p = .083$. There was also a significant interaction between hint and time limit, $\beta = -.033$, $p = .036$, and a marginally significant interaction between hint and number of unique candidates, $\beta = .23$, $p = .09$. Generating more candidates was helpful if a hint was seen but had the opposite effect in the case where a hint was not seen. The hint was also less effective at the longer time intervals.

Including impasse as a factor in the regression does not explain any more of the variance as the impasse factor as well as all interactions involving the impasse factor did not approach significance. It seems that the combination of time limit and number of unique candidates explains the surprising impasse results shown in Figure 17. The hint by time limit interaction can be explained by fixation. Generating four candidates in the 15 s time limit leaves less time for repetition and the subsequent increase in fixation than generating the same four candidates in 30 s. The increase in fixation impacts the effectiveness of the hint which explains the separation of the three lines corresponding to different time limits in the hint condition in Figure 23. Generating a large number of candidates for a particular problem can be seen as exploring the problem better or developing a better representation of the problem and things that are related to it. This exploration concept can help explain the interaction between hint and number of candidates.

When a participant has explored the problem a great deal without finding the answer, it is less likely that the second attempt will lead to the answer since the search space for the problem has been exhausted. However, when a hint is presented then exploring the problem makes it more likely that the hint becomes incorporated into problem solving presumably because the better representation of the problem makes it more likely that the similarity between the unsolved problem and the hint will be noticed by an open goal mechanism. The complementary concepts of fixation and exploration can therefore help in understanding the results presented in Figures 17 and 23.

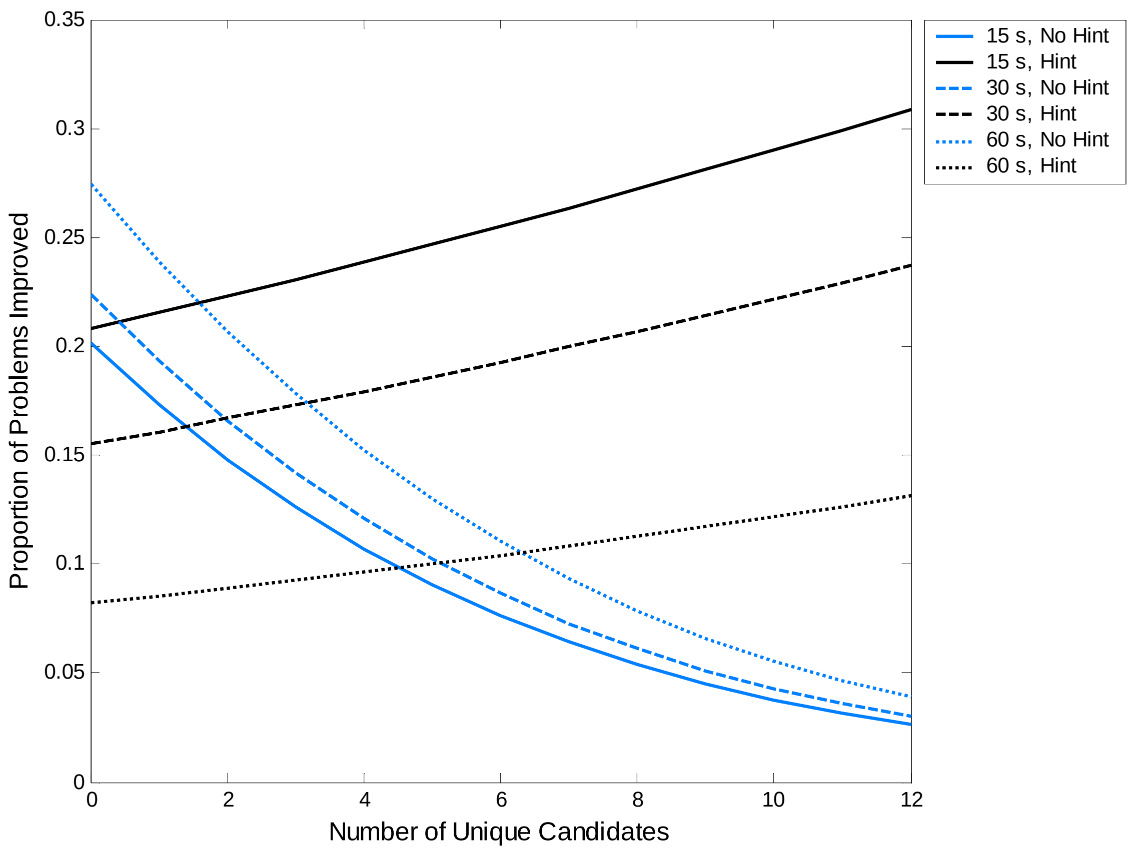


Figure 23. Results of logistic regression used in explaining fixation and impasses

*Feelings of Warmth*

Participants were asked to rate their feeling of warmth on a scale of 1-5 for problems that they left unsolved. The warmth ratings for two participants were not collected due to technical difficulties. The ratings were analyzed to see if they were indicative of improvement or hint effectiveness. As can be seen in Figure 24 the feeling of warmth measure was not indicative of improvement or hint effectiveness. Participants generally rated problems with a low feeling of warmth regardless of whether they were eventually able to solve the problem. This result indicates that participants were generally unable to assess their proximity to the solution for the RAT problems.

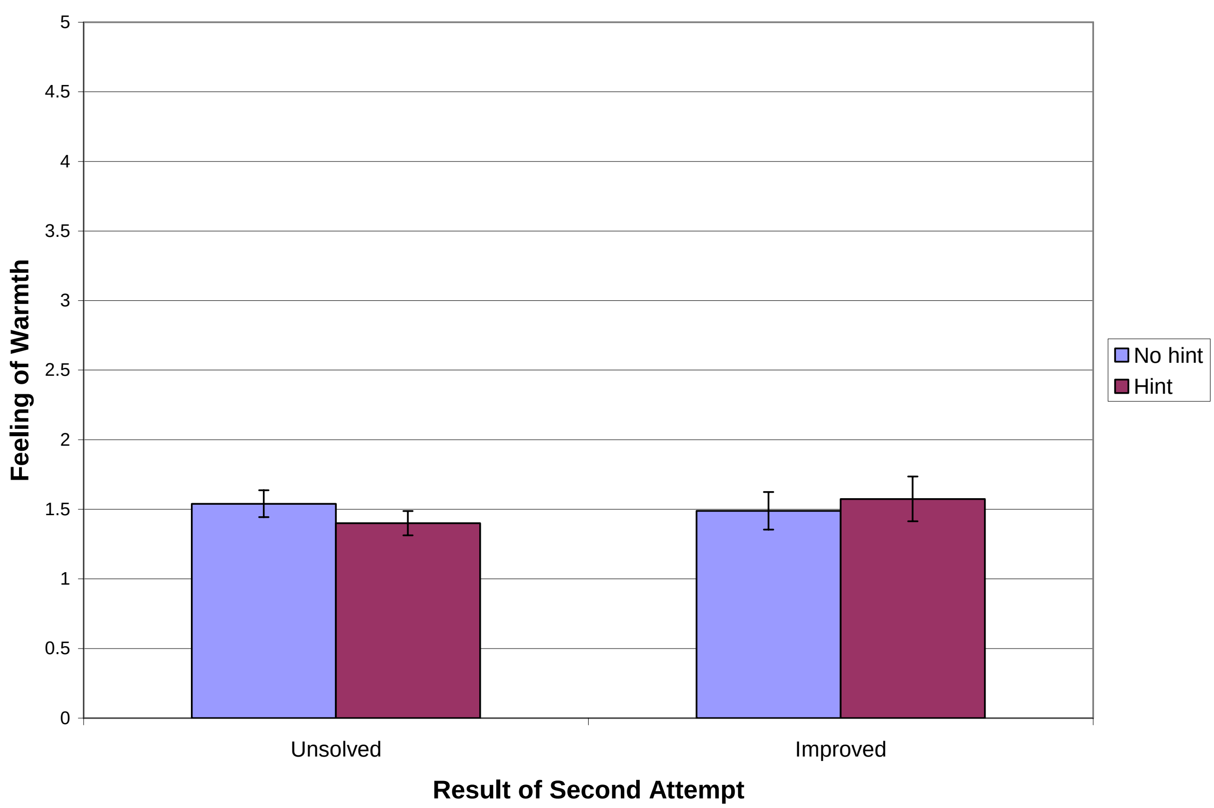


Figure 24. Feeling of warmth ratings on problems unsolved after first attempt

*Free Recall of RAT Problems*

Data from the free recall task at the end of the study are presented in Figure 25. The proportion of improved, unsolved, and correct problems were analyzed by collapsing across the three time limits. The analysis collapsed across participants because less than half of the participants improved on problems at all three time limits. The means show the same patterns that were observed in Study 1B where improved problems are recalled at a higher rate than unsolved and solved problems, and unsolved problems were recalled higher than solved problems, but none of the means were significantly different.

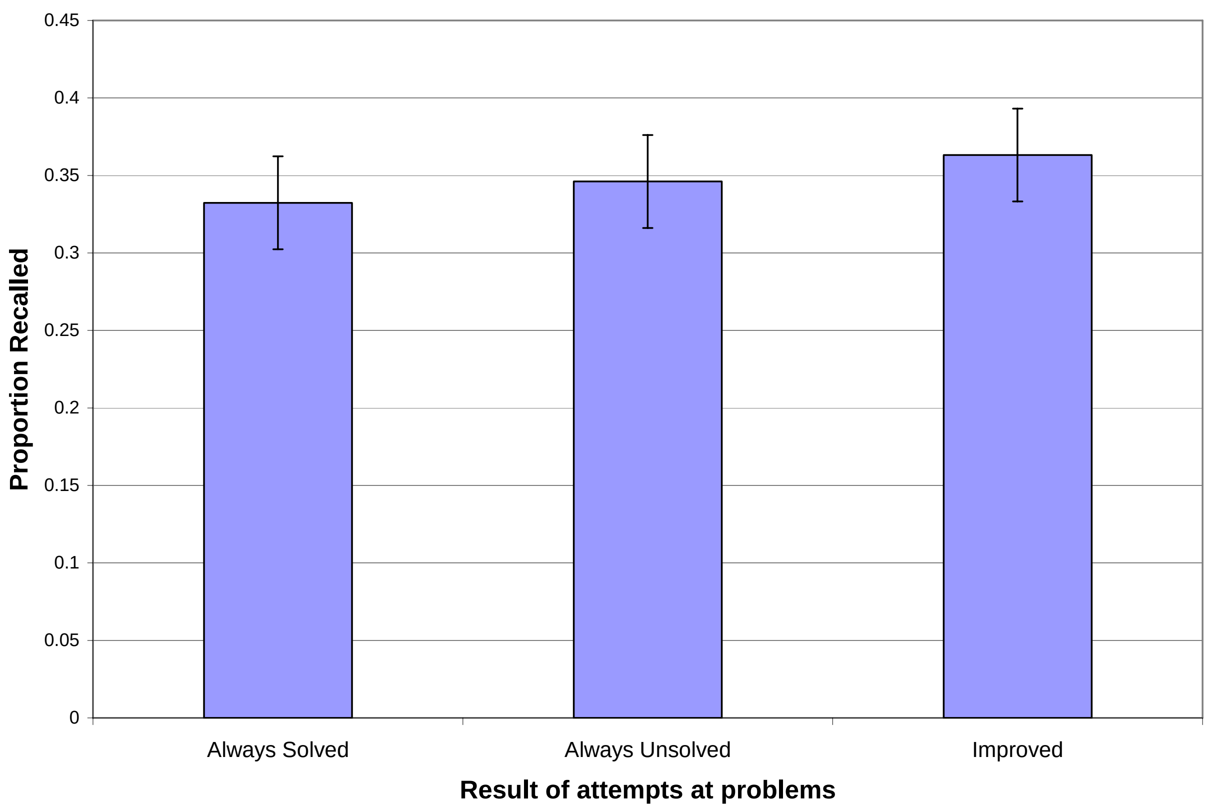


Figure 25. Proportion of problems recalled by solution status

There were not enough of each type of problem recalled at each time limit to break these results down by time limit, but it was possible to examine the impact of the time limit by combining the two longer time limits of 30 s and 60 s. These data are presented in Figure 26. The pattern of means from the 15 s time limit is different from the other two time limits. The longer time limits resemble the pattern of recall results from Study 1B presented in Figure 7 where a 30 second time limit was used. The interaction between time limit and solution status was marginally significant, $F(2,44) = 2.96$, $p = .062$.

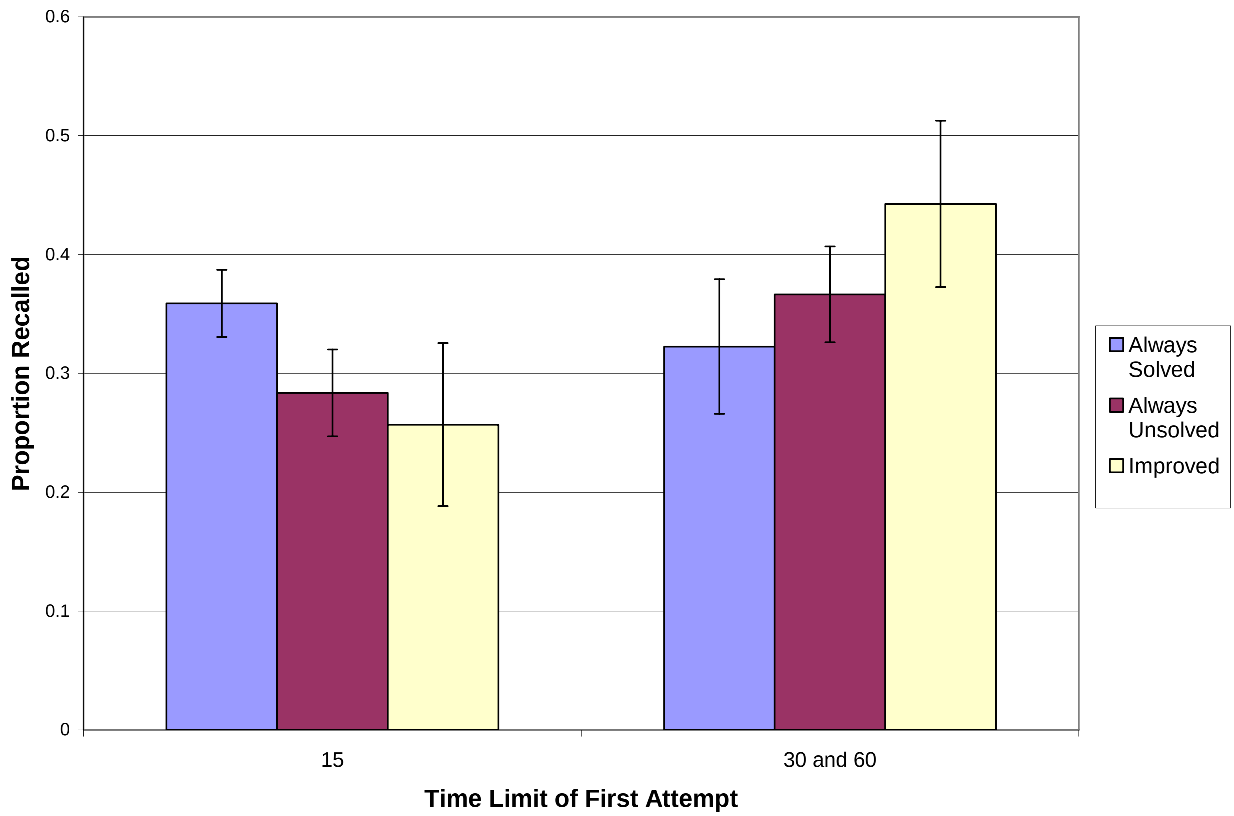


Figure 26. Proportion of each class of problems recalled

The amount of time actually spent on the problems for each class of problems at the two levels of time limit is shown in Figure 27. The increased time spent on solved problems does not translate into an increase in proportion of problems recalled, but for unsolved and improved problems the increased amount of time does affect the proportion of problems recalled. In order to rule out an explanation based just on time spent on the problem a regression analysis was performed. Even after the variance by time spent on the problem is accounted for, there is still a marginally significant interaction between time limit and solution status, $p = .088$.

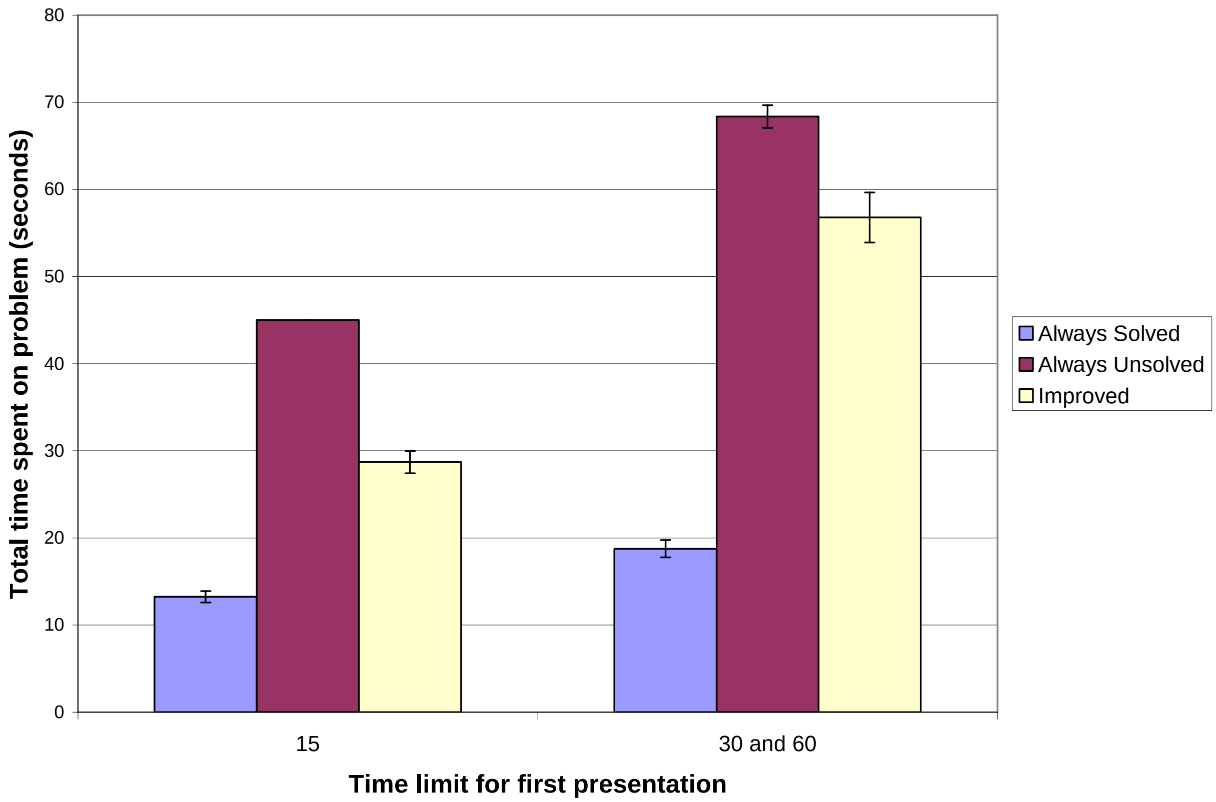


Figure 27. Total time spent on each class of problems

Recall for initially unsolved problems was also examined in terms of whether an impasse was reached in problem solving. There was no difference in the proportion of impasse and non-impasse problems that were recalled.

*4.1.3 Discussion*

A number of the results are consistent with what was found in the previous studies. While there was no significant effect of hint at the 30 s time limit, the pattern of means was in the correct direction. There may have been some effect of either the protocols or the varying time limit which changed the effect size. However, there is still a significant effect of hint on solution time for problems which were solved on the second attempt as was found in Study 1B. The feeling of warmth measure was not related to hint effectiveness as was found in Study 2. The results from the free recall task also replicate the finding from Study 1B that improved problems are more likely to be recalled than solved problems at least at the longer time limits. The interaction with time limit is discussed more below.

One purpose of this study was to examine what role impasses play in making use of problem relevant information presented in this case in the form of implicit hints embedded in the lexical decision task. The expectation was that reaching an impasse would make participants more likely to make use of the information in the hint. One reason for this expectation is that reaching an impasse means that the representation of the problem was likely to be better developed than on a problem where there was an interruption early in the problem solving. Another reason is that it has been found that problems in which an impasse has been reached were more likely to be recalled than problems where participants had not been allowed to reach an impasse (Patalano & Seifert, 1994; Seifert et al., 1995; Seifert & Patalano, 1991). It seems reasonable that if impasse problems are more accessible then the higher accessibility or activation of these problems may lead to their recall when useful information is encountered.

The finding that the hint was only effective for non-impasse problems was surprising given this expectation. There are a number of possible reasons for this interaction between hint and impasse. One is that problem solving changed qualitatively after the impasse in such a way that made the hint less effective. The results of the study indicate that participants made no quantitative change in problem solving as they generated the same amount of new and old candidates on their second attempt at impasse and non-impasse problems.

Other work on impasse problem solving has suggested that processes of constraint relaxation and chunk decomposition are used to overcome impasses encountered in insight problem solving (Knoblich, Ohlsson, Haider, & Rhenius, 1999). A qualitative change such as relaxing some constraints or trying more distantly related candidates may have followed the impasse. This kind of change would mean that there were potentially many more available associates and so the hint would not have effectively increased the likelihood of retrieving the answer given the increased number of other candidates it must now compete with.

Another explanation for the lack of an effect of hint in the case where an impasse was reached is that participants are giving up when they reach the impasse. However, this is not likely to be the case since there were no significant changes in the number of candidates generated on the second attempt at impasse and non-impasse problems.

Impasse problems were not differentially recalled more than non-impasse problems. It is the case that across studies there is an advantage for unsolved over solved problems which is in line with other findings (Patalano & Seifert, 1994; Seifert et al., 1995; Seifert & Patalano, 1991), but the claim that reaching an impasse on a problem leads to higher recall does not seem to hold for all types of problems. Seifert and colleagues argued that their results support the Zeigarnik effect (Zeigarnik, 1927/1938) that interrupted tasks were recalled more than uninterrupted tasks. However, it is not clear that interrupting participants after they have reached an impasse is more in line with the original Zeigarnik effect than interrupting participants before they have reached an impasse while they are in the midst of problem solving. Given that in the original studies by Zeigarnik a participant was interrupted at a time when “the subject was most engrossed in his work” (Zeigarnik, 1927/1938, p. 303), it seems that the original Zeigarnik effect would apply more to the non-impasse RAT problems in this study. It may be that these problems are the ones which are at a heightened level of accessibility and become associated with the hint when it is presented.

The impasse results may be related to fixation. The results from the logistic regression analysis support this explanation. The results show that the concepts of fixation and exploration explain the surprising finding on insights. As work on a problem progresses, the problem solver explores more of the problem space and develops a better representation of the problem. When an impasse is encountered, the problem solver is unable to generate a new idea for how to proceed. If work on the problem continues, then a significant amount of fixation will be built up because the problem solver has nothing to do but repeat prior failed attempts at the problem.

Using this line of reasoning, one explanation for the surprising impasse results is that participants were forced to continue working on a problem after an impasse had been reached. This situation led to a significant increase in the amount of fixation which impacted the effectiveness of the hint. If participants had not been forced to work during the impasse then they might have disengaged from the problem which would have prevented most of the buildup in fixation. The results presented here begin to unpack the processes that are involved in impasses including those of problem exploration and fixation. Given these results, the time at which open goals should have the greatest impact would be just after an impasse has been reached. If the problem is disengaged from at this point, then there will be some benefit from problem exploration without a large amount of fixation being built up through the repetition of failed problem attempts.

The results of this study did confirm that a significant amount of fixation did occur, and so fixation is likely to be one of the main sources of difficulty for these problems. This result is consistent with other work on RAT problems (Smith & Blankenship, 1991; Wiley, 1998). Fixation will play an important role in the model of RAT problem solving discussed below.

# Chapter 5

# Modeling the Effects of Open Goals

## 5.1 Purpose of Model

In the interpretation of the results of Study 1B it was hypothesized that a temporary activation increase associated with seeing the hint could not fully explain the results that were found. To support this argument, a cognitive model of performance in this study was constructed where the only effect of the hint was exactly this temporary increase in activation. It was predicted that this model could account for the results of Study 1B in every condition except the case where a hint was shown for previously unsolved problems.

Fixation on distractors is the reason why such a model can not account for this one condition. Hints should be less effective when there are competing distractors that have been activated by a relatively recent problem-solving attempt. However, this was not the result found in Study 1B. Further support for this interpretation was found in Study 3 where fixation on distractors was shown directly through the examination of repeated candidates in the verbal protocols.

One of the purposes of producing a model of problem solving in this one study is to verify that the effects of fixation do in fact account for the increased difficulty of the problems that were left unsolved. The idea is basically that if the model is built without any kind of open goal mechanism then it should fail to account for the amount of improvement seen in the previously unsolved problems with a hint. However, the model should be able to account for the effect of the hint in the case of unseen problems since this was caused by a temporary increase in the hints' activations due to processing the word in the lexical decision task.

In some sense, the model is being setup to fail, but in this case the failure is informative and provides support for the idea that there needs to be an open goal mechanism to account for the results presented in this dissertation. Furthermore, another reason for constructing the model is to have a working model that can serve as a test bed for any proposed open goal mechanism. Modifying the model to include this mechanism should allow the results in Study 1B to be fully accounted for.

The model was built using the ACT-R architecture (Anderson et al., 2004; Anderson & Lebiere, 1998). An overview of the aspects of the architecture that are the most relevant for the current model is presnted in the next section before the details of the model are presented.

### 5.2 Memory Retrieval

Retrieval in ACT-R is determined by the relative activation levels of all chunks that match the retrieval request. The activation level of a chunk is determined by adding together three components: base level activation, spreading activation, and noise. The sum of these three values determined the chunk's activation value ($A_i$), as shown in Equation 1. The noise value ($\varepsilon$) is drawn from a logistic distribution with a mean of zero and a standard deviation controlled by a parameter in the model. The base level activation ($B_i$) is a value determined by the frequency of use of a chunk since its creation. The base level activation is determined by Equation 2 where $t_j$ is the time of the $j^{th}$ time the chunk was retrieved or encountered and d is the decay rate which is fixed to a value of .5 in ACT-R. The spreading activation component of the equation consists of two components. Each element of the current goal spreads activation in memory equally so that

$W_j$ would be equal to 1/n where n is the number of elements or slots for holding chunks in the current goal. $S_{ji}$ is a measure of the strength of association between the $j^{th}$ element of the current goal and chunk i which is the chunk for which activation is being calculated.

$$A_i = B_i + \sum_j W_j S_{ji} + \varepsilon \tag{1}$$

$$B_i = \ln(\sum_{j=1}^{n} t_j^{-d}) \tag{2}$$

In general, these equations mean that a chunk's probability of being retrieved successfully depends on how recently and frequently it has been used as well as its strength of association to elements within the current goal. In ACT-R, the chunk with the highest activation value at the time the retrieval request is made is the chunk that is retrieved. Activation values vary because of the passage of time (decay), changes in the current goal, and noise. The probability of retrieving a particular chunk i given a set of chunks j is given by Equation 3.. This means that the probability (P) that a particular chunk has the highest activation and is retrieved is a function of the noise parameter (s) and the activation values excluding noise (m) of the chunk and all of the other chunks that match the retrieval request.

$$P(i) = \frac{e^{m_i / s}}{\sum_j e^{m_j / s}} \tag{3}$$

For RAT problems and the presentation of a hint it is the base level activation from Equation 2 that will play the largest role since this is where the presentation of the hint will have its effect. As an example of how these equations can help to explain the results of Study 1B an example RAT problem-solving case will be presented. For this example, it will be assumed that there are three distractor words that are retrieved by the problem solver during the initial attempt at the problem. One of these is retrieved three times, one twice, and the other once. The activations of the chunks which represent those distractor words are presented in Figure 28 and are determined by Equation 2. In this figure the activation values for the three distractors become activated during problem solving as can be seen on the far left of the graph as time increases from left to right. Once the problem solver has moved on these chunks receive no further increase in activation and begin to decay according to Equation 2.

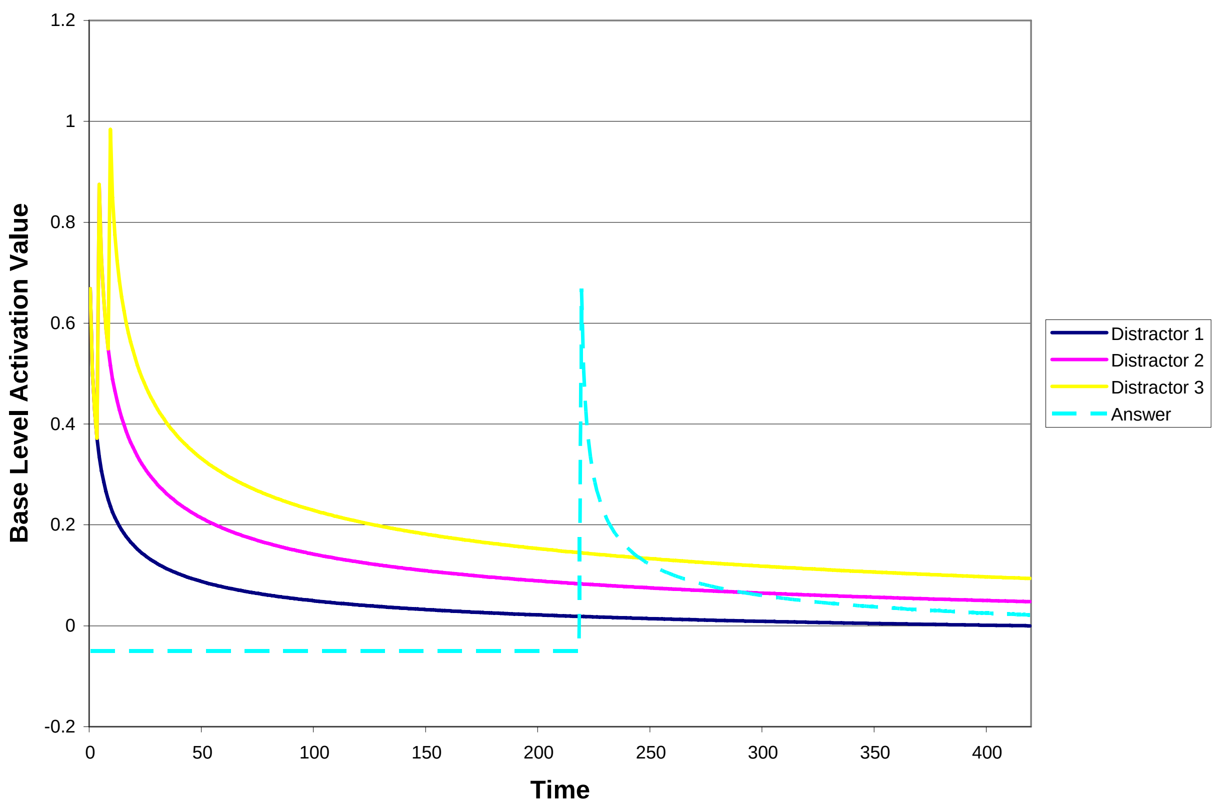


Figure 28. Activation levels of words for example RAT problem

The answer to the RAT problem is represented by the line that starts out and stays slightly below zero on the left side of the graph. The answer is never retrieved and so stays at a low level of activation until it is processed in the lexical decision task. By the time the problem is encountered again near the right hand side of the graph the answer is still at a relative disadvantage when compared to its distractors. If the problem had not been seen initially, then the distractor words would be at their initial base level of activation just as the answer word was at time zero. The activation values for these four words can be converted into the probability of retrieving the answer according to Equation 3.

The probability of retrieving the answer using the values in Figure 28 is displayed in Figure 29 for four different conditions. In the unsolved-no hint condition the distractors shown above are activated through retrieval, but the answer never increases because it is never presented as a hint. In the unseen-no hint condition the answer and all of the distractors are at equal baseline levels, and so all have an equal chance of being retrieved. In the hint condition, the problem can either be a new previously unseen problem or a problem that was previously unsolved. In the case where the problem has not been seen the distractor words will not be good competitors since their activation levels have remained at their baseline levels. This leads to the ordering of the conditions at the right side of the graph where the most successful condition is the unseen-hint condition followed by the unseen-no hint condition, and then the unsolved-hint and the unsolved-no hint condition. These values all represent the probabilities for a single retrieval, and there are multiple retrievals during a problem-solving attempt. However, this example illustrates the key mechanisms in the model.

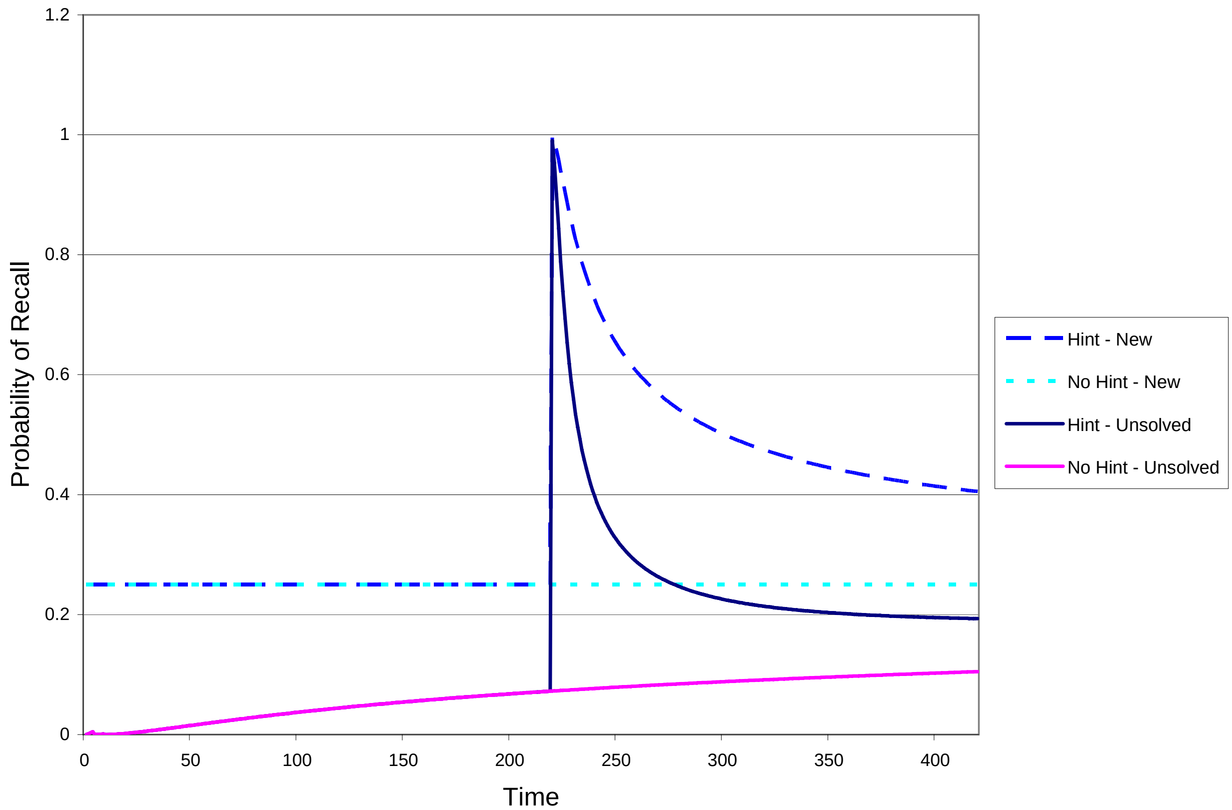


Figure 29. Probability of retrieving the answer for example problem

## 5.3 Model Details

RAT problems are not very complex in the sense that they do not require sophisticated problem-solving methods to solve. Based on an analysis of the protocols from Study 3, the most common way of solving these problems is a simple generate-and-test method. There also seems to be a small percent of the problems that were solved simply by reading the words and the answer just popped into mind. This last method may not be a different method at all as it may just be that the first cycle of the generate-and-test algorithm was successful. It is hard to say whether it is the same or not because in these cases participants did not verbalize much at all beyond the answer word.

A generate-and-test algorithm for solving RAT problems was proposed by Smith (1995a):

1. Select a test word.

2. Retrieve an associate of that test word.

3. Select another test word.

4. If the word is an associate of the second test word, then go to step 5, else go to step 1.

5. Select remaining test word.

6. If the word is an associate of the third test word, then done, else go to step 1.

This algorithm is roughly consistent with what was found in the protocols, but there are a number of unspecified details as well as other modifications that are needed in order to produce a model of RAT problem solving.

First, the problems that were used in these studies were compound remote associates that have a particular type of relationship between the answer and the words given (Bowden & Jung-Beeman, 2003b). This means that people could try to generate any kind of associate to the word or they could specifically try to retrieve a compound word. Both types of retrieval seem to take place in the protocols.

The basic structure of the model for solving RAT problems is a series of repeated retrieval attempts using one of the three strategies. If a compound word can be retrieved, then it can be tested against the other words of the problem. During retrieval all three words of the problem serve as sources of activation. This model is able to solve RAT problems, and it also demonstrates a certain degree of fixation. The ACT-R theory provides a way for the model to mark each compound word that it retrieves so that it will not be recalled again. However only a fixed number of these chunks can be marked and the marks decay with time. This situation leads to the repeated recall of words related to the RAT problem but which are not the answer. Each time a word is recalled it receives an increase in its base level activation which decays with time. The repeated recall of related but incorrect words which will be referred to as distractor words leads to the situation where the most likely item to be recalled is from the set of distractor words which are exactly the words which will not help solve the problem. This type of fixation in RAT problems is very similar to what was found by Smith and Blankenship (1991).

Taking a break from working on the problem will allow these distractor items to decay with time. In ACT-R the chunk with the highest activation level that matches the retrieval request will be recalled. However, noise in the activation levels leads to the situation where the probability of recall of a particular chunk is based on its activation level relative to the activation of other chunks that match the request as well as the value of the noise parameter. This means that as the distractor words decay the probability of the answer being recalled increases. These mechanisms should account for the percent of unsolved problems that were solved the second time in Studies 1A and 1B when the answer was not presented as well as for the incubation and fixation effects found by Smith and Blankenship (1991).

The first step of the process given above implies that one word is selected and used to guide selection. The selection of this word could be random or there could be some kind of strategy used to guide selection. The most commonly used strategies found in the protocol study were to use the first word, use the least or most frequent word, or to use the word that is easiest or hardest to find compounds for. Retrieval of an associate word using any of these strategies would also be influenced by the other two words that appear in the problem. This aspect of problem solving is consistent with theories of spreading activation in memory.

The actual retrieval of associates is affected by the structure of declarative memory. The protocols show that participants seem to be accessing an intermediate “meaning” or conceptual representation as they generated candidate answers. For example, given the problem “iron”, “engine”, and “shovel” it was found that candidates retrieved often corresponded to some kind of industrial or mechanical concept such as car engines, gears, or tools. The answer is steam which is related to the dominant concept through steam engine and steam shovel. However a participant selecting “iron” to start from may not have generated steam iron given that this concept is not as related to the industrial/mechanical meanings brought to mind.

The basics of this kind of memory structure were used in the model presented here. Essentially each word was linked to one or more meanings which were in turn linked to one or more word or compound words through associative links. A typical example of this kind of representation is shown in Figure 30. The model’s representation is not as complex as it could be. In this model, most words only have one meaning. The only exception to this is that the answer words to the RAT problems are associated with three meanings since they must be associated with the meanings of three compound words in order to be answers to RAT problems. This simplification should not have a large impact on the model’s performance, and the consequences of this simplification are discussed below.

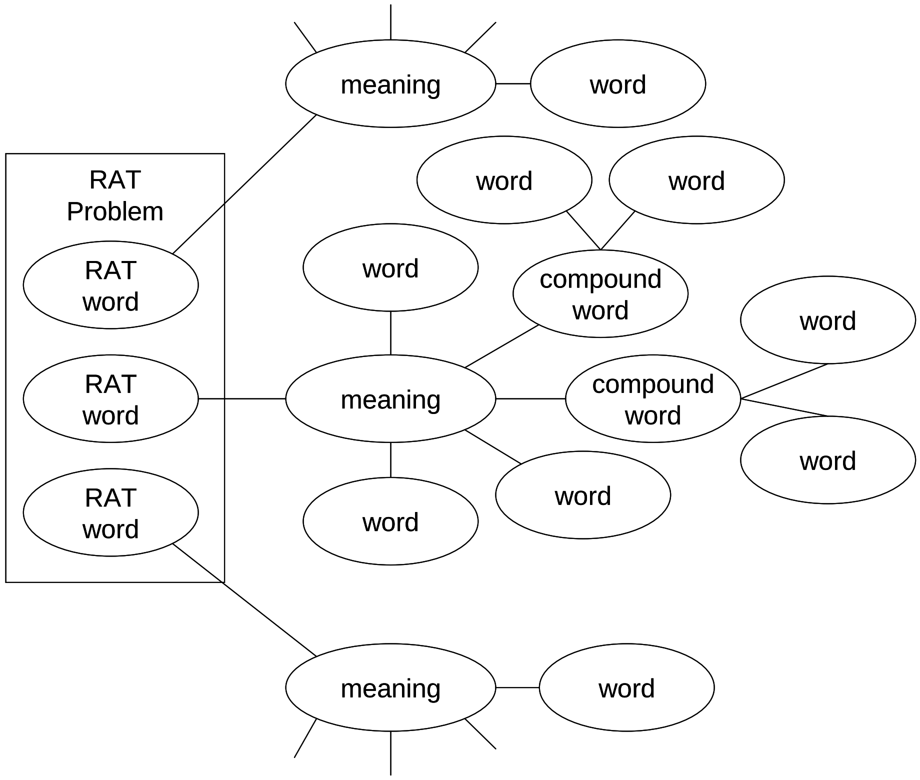


Figure 30. Example of the structure of declarative memory.

The model begins by encoding the three words that make up the problem. The next step is to select a word and access its meaning. The model currently chooses randomly between the three words, but it would also be possible to model other strategies such as starting with the least frequent word. After the semantic meaning of the word has been accessed, the model either attempts to retrieve a related word or a related compound word. During this retrieval activation is spreading from the three words in the RAT problem as well as the current meaning that was just selected. This type of retrieval is supported by the protocols as participants seem to think about a particular semantic concept associated with one or more words and then retrieve words related to that concept. If a candidate word or compound word is retrieved it is tested against the other words in the problem. If no word or compound word is retrieved then the model selects a problem word and begins the retrieval process over again.

When testing a word, the model makes a series of retrieval requests in order to search for a compound word in memory which corresponds to the candidate word placed before or after each of the problem words. If a candidate word is found then that word is entered. All words are assumed to be five letters in length as the model types the response and presses the “enter” key. The experiment is presented to the model just as the experiment was presented to participants with the model having exactly 30 seconds in which to respond. Each problem is followed by feedback which the model encodes as part of the problem-solving goal that eventually ends up in declarative memory when the problem has been taken away and the model moves on to the next task.

The model for the lexical decision task is straightforward as the model encodes the string presented and makes a retrieval request to determine if the string is a word or not. The hints are presented during the lexical decision task. The retrieval request for the hint word increases the base level activation of the word temporarily as it will eventually decay back to a relatively stable baseline activation. This temporary increase is the only effect of the hint in the model.

### 5.4 Model Fit

As discussed above, the general strategy for the modeling work is to develop a model which can account for most of the data observed in Studies 1A and 1B, but to test whether it will be able to account for the entire amount of the increase in percent of problems solved without some sore of open goal mechanism.

The data collected from the studies presented here provide some constraints for fitting the model. The key parameters affecting solution of the RAT problems are the initial base levels of the answers, the retrieval threshold, the decay rate, the number of different words that match a retrieval request, the level of noise in activation values, and the particular retrieval strategy being used. Most of these parameters can be fixed or estimated. Three retrieval strategies have been identified, and they can be incorporated into the model with different strategies being used for different participants. The base levels of the words to be recalled can be assumed to form some distribution of subthreshold activation values. The decay rate is usually a fixed parameter in ACT-R models. The number of different words that match a retrieval request is determined by the number of different compound words that contain a given word, and the number of such words can be determined from a database like WordNet (Fellbaum, 1998). This leaves essentially the retrieval threshold and noise parameters as the two parameters to be estimated from the data collected. The model fitting was to fit the data using these parameters to three of the four conditions in Study 1B. The one condition excluded will be the unsolved problems for which the answer was presented. By fitting the model on the other three conditions, the model can then predict performance on this one key condition. It was predicted that the model will be unable to account for the boost in performance in this one condition because this was the one condition where open goals were influencing what information became incorporated into problem solving.

Using this procedure it was possible to produce a fit to the data as shown in Figure 31. After adjusting parameters to fit every condition but the unsolved-hint condition, the model predicts that participants will be less successful in the unsolved-hint condition than they actually were. These results are in line with what was expected and demonstrate the need for some kind of open goal mechanism in order to fully account for the results of the studies presented in this dissertation.

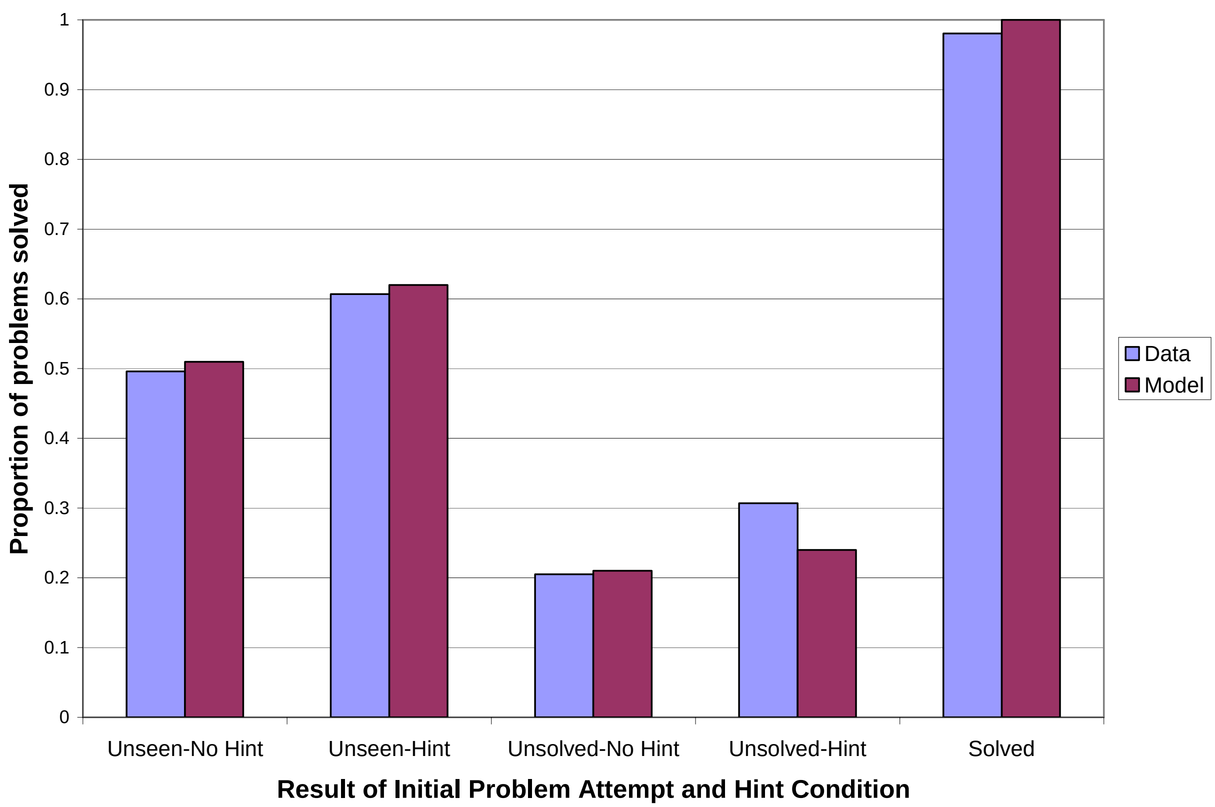


Figure 31. Model fit to the data

## 5.5 Discussion

The results of the effect of the hint in the unseen and unsolved conditions were as expected. The hint was more effective if the problem had not yet been seen than in the case where the problem had been left unsolved. These results demonstrate that there is a mechanism unaccounted for in the model. Furthermore, the results also indicate that fixation is a feasible explanation for the increased difficulty of previously unsolved RAT problems, and the modeling results support that interpretation of the results from Study 1B.

Before the model can serve as a test bed for any new mechanism it will be important to further compare the model to the data collected in the protocol study. For example, it would be important to see if the number of retrievals the model is making in a problem solving attempt is similar to the number of retrievals that the human participants are making which can be estimated from the verbal protocols. Additionally, solution time results were not collected from the model, but this data can be collected from the model and compared to the data from the studies. Also, the declarative representation is very basic in the sense that it is likely missing some of the complexities of similarity relations found in human participants. Future modeling work will focus on determining which of these aspects of the model need to be changed before the model can serve as a test bed for potential open goal mechanisms.

Another aspect of RAT problem solving that can be evaluated with a working model is the relationship to working memory. It was originally hypothesized that working memory would be related to RAT problem solving because of the fact that individual differences in working memory capacity have been related to the amount of activation that can be spread in ACT-R (Daily et al., 2001; Lovett et al., 2001; Lovett et al., 1999). Essentially the working memory parameter being manipulated is related to the W constant in Equation 1. More activation to spread would mean that more distantly related items could be recalled which would lead to greater success at solving RAT problems. Unfortunately, the current model needs to be further assessed before exploring alternative mechanisms and the relationship with working memory.

The next step in the development of this model is to explore modifications to the declarative and procedural representations used in the task to examine how they impact problem solving and the relative effectiveness of the hint. Once the model matches the all of the aspects of the data discussed above it should be possible to use it to test out any proposed open goal mechanism. The code for the model can be found in Appendix II.

# Chapter 6

# General Discussion and Conclusions

## 6.1 General Discussion

The idea that open goals influence the acquisition of problem relevant information was based on ideas and results from the insight and incubation literature (Kaplan, 1989; Seifert et al., 1995). The results presented in this dissertation provide the beginnings of a theory for how an open goal mechanism influences information acquisition. This theory provides a new perspective on a variety of topics related to problem solving including insight, incubation, transfer, and creativity. In addition, influences from open goals seem to be a more general phenomenon that extends to areas of attention, task switching, and memory. Consideration of the role that open goals play in these other areas should provide further constraints on the development of the theory. The resulting theory should provide insight into a number of areas of cognition while at the same time contributing to the goal of a unified theory as it relates phenomena from the problem-solving literature to other areas of cognition.

### *6.1.2 Insight, Incubation, and Creativity*

Much of the original motivation for this work came from a desire to understand how insight happens including how people are able to generate new representations during problem solving. The open goals theory being developed here does not explain all insight nor does it necessarily compete with other theories of insight. It is more of a complementary idea that helps to bring theories of insight problem solving in touch with lower level processes such as attention and memory.

Some theories of insight problem solving have focused on search and the sources of difficulty in insight problems (Kaplan & Simon, 1990; Kershaw & Ohlsson, 2004). The influence of open goals on acquiring problem relevant information fits with aspects of these theories by explaining when new information may enter the search process and possibilities for how new information may help people overcome the various types of difficulties encountered when solving an insight problem.

For instance, it was found that hints about a new kind of representation for the mutilated checkerboard problem were only effective when people when people had given up on their current representation and begun to search for a new one (Kaplan & Simon, 1990). This problem involves trying to place dominoes on a modified checkerboard so that the entire board is covered, but the tiling is impossible because of the way in which the board has been modified. Noticing the parity of the colored squares that dominoes covered was essential in producing the solution because two of the same colored squares had been removed from the board in order to make a tiling impossible. People who are still attempting to find a tiling solution and mention parity do not immediately solve the problem by proving it impossible. However, people who had given up on tiling the board and were instead trying to prove it impossible solved the problem within a few minutes of mentioning parity. The representation of the goal of problem solving in this case determined whether participants incorporated parity into their problem solving after they had noticed it.

While the results of the mutilated checkerboard study do not necessarily indicate any effects of open goals, they do indicate that the representation of the goal affects what information is incorporated into the problem-solving process. It could be the case that while exploring alternative representations of the problem that there were numerous other goals set up to explore one idea or another. These are subgoals of the original goal of proving the problem impossible. If open goals such as the original goal continue to influence information acquisition, then it would be predicted that noticing parity while operating under any of these subgoals would lead to parity being incorporated into problem solving even if parity is not relevant to the current subgoal being used to explore the current representation. It is not possible to say with certainty that this was the case given the data presented by Kaplan and Simon (1990), but this is one example of how the influence of open goals complements and might extend current theories of insight.

Noticing information relevant to a previous subgoal or subproblem could explain the opportunistic deviations in engineering design problem solving discussed earlier (Guindon, 1990; Visser, 1990, 1996). These results highlight how open goals play a role while still engaged in problem solving. However, it seems even more likely that open goals are even more relevant when one considers information acquired while problem solving is suspended. Research on incubation has generally found that taking a break from problems often helps. Open goals are an explanation of interactive incubation effects. While disengaged from the problem, the problem solver may encounter relevant information which becomes incorporated into problem solving.

There is also evidence that the simple passage of time produces what has been termed an autonomous incubation effect (e.g., Kaplan, 1989; Smith & Blankenship, 1991). One of the primary explanations for autonomous incubation effects is that interference from previous ideas associated with the original problem-solving episode decays over time (Kaplan, 1989; Smith & Blankenship, 1991). Kaplan also suggests that priming may lead to what has been termed an interactive incubation effect. However, the only details offered on how such a priming mechanism would operate are that, “The act of priming corresponds to increasing the activation of a node in LTM, and, more importantly, to strengthening the links to that node. Stronger links increase the probability that the concept will be retrieved in the future.” (Kaplan, 1989, p. 94).

Within an architecture like ACT-R (Anderson et al., 2004), priming might be accounted for by a temporary increase in a chunk’s activation level like that shown in the model presented above. However, this kind of account would not explain the long-term priming across hours and days that Kaplan (1989) observed. It also did not fully explain the improvement in solution rate associated with the condition of where hints had been presented for previously unsolved problems in Study 1B. Kaplan’s explanation of increasing links or associations to words indiscriminately increases the chunk’s associations to everything else. While this would increase the chances that the problems solver would retrieve that chunk in problem solving, it only does so if one assumes that only problem relevant information becomes primed. If every piece of information that a person encounters is primed in this manner, then there will be ever increasing strengths of association between many chunks in memory. In order to avoid problems under such a theory one must assume that these associations also decay over time, which would seem to lead to the same problems as a temporary increase in activation.

A theory of long-term priming during incubation needs some way to discriminate relevant information from irrelevant information. A lasting influence of open goals provides the beginnings of a mechanism that would provide such discrimination. Modifying Kaplan’s explanation of priming above to say that only the association between the open goal and the relevant primed chunk becomes strengthened is one potential way in which open goals could exert their influence during incubation. It is still necessary to determine when a chunk is selected for such a strengthening process, and this will be discussed below.

Generating new representations and ideas while solving a problem is also related to the study of creativity. The work presented here provides the beginnings of a theory that could explain how some creative ideas are generated. People are generally able to map helpful information over to a new problem situation, but the difficulty comes in noticing when the information is relevant (Gick & Holyoak, 1980). Some ideas about creativity and insight involve just these kinds of mapping processes (Langley & Jones, 1988). Other theories involve combining various ideas and concepts into new variations (Campbell, 1960; Simonton, 1999). The idea that open goals direct the noticing of these kinds of relationships could go a long way in furthering our understanding of how creative ideas come about without positing a large amount of nonconscious processes. For the most part the processes operating are essentially processes of attending to things in the environment and making associations in memory which are known to occur outside of consciousness anyway.

### *6.1.3 Transfer*

Noticing relevant information and mapping it onto the current problem may be creative in some instances, but it is generally referred to as transfer. As noted earlier, incubation may be a special instance of transfer where information that is noticed during incubation becomes incorporated into problem solving.

Christensen and Schunn (2005) present data which support this view of incubation and transfer. However, they also note that Gick & Holyoak (1980) did present one study where they presented the source material for the analogy during problem-solving sessions with the target problem. Gick and Holyoak found no benefit of presenting the source material in this order when compared to presenting the material before the target problem was attempted. These results conflict with those of Christensen and Schunn which suggests that analogical transfer should be higher when the source information is presented between problem-solving attempts. They note this discrepancy but merely suggest that the original Gick and Holyoak results "should be retested using a variation of this design in a larger study" (Christensen & Schunn, 2005, p. 217).

If one examines this discrepancy with the idea that open goals are necessary for this noticing and mapping to take place then the difference between the experiments explains the difference in results. In the study by Gick and Holyoak participants were asked to generate as many solutions to the problem as they could. Throughout the multiple studies they present in their paper, participants average at least two alternative solutions beyond the actual analogical solution that the authors were specifically looking for. Given this situation, it seems likely that participants had generated one or more solution for the problem already by the time the relevant source material was presented 10 minutes into the work on the target problem. In this situation, many participants may not have an open goal for the problem. To them the problem is done and they are just continuing to work because they are in an experiment and there is nothing else to do. There is no unsolved problem or open goal for them because the experiment is such that they are never told that their potential solutions are inadequate.

Christensen and Schunn used insight word problems with a clearly defined solution. In this situation, participants have temporarily abandoned the problem because they do not know how to proceed in solving it. An open goal has been left, and the authors found that there was a good deal of analogical transfer in this situation. By introducing the open goal construct it becomes possible to understand these apparently conflicting results.

Another related result in the analogical transfer literature is that inducing the formation of schemas increases the chances of analogical transfer (Gick & Holyoak, 1983). Schemas are representations that represent similarities in the structure of problems by abstracting away from the surface form of the problems. By representing only the structural similarities this kind of structure allows people to recognize instances where the same structural similarities are present and knowledge could be transferred successfully. To the extent that the representation of problems associated with open goals includes the structural aspects of the problem, then matching new information to open problems could be similar to using schemas to transfer knowledge. Working on a problem long enough to develop a representation of the structure of the problem would then promote the recognition of relevant information. In addition, experts, who are likely to represent and remember problems in their domain through their deep structural properties, may be more likely to notice problem relevant information when it is encountered.

### *6.1.4 Mechanism*

One of the properties of the theory being developed here is that it links problem solving with other areas of cognition such as memory and attention. These other areas will help to constrain the development of the theory, and will contribute to the generality and parsimony of the theory as it will help to understand results in a variety of areas. Within the area of problem solving, open goals have been linked to the effectiveness of hints, insight, incubation, and knowledge transfer. The results presented in this dissertation demonstrate that open goals can influence the acquisition and use of problem relevant information. A model of the results from study 1B was implemented to test out the ideas of fixation and hint effectiveness that served as the basis for the interpretation of the results for that study. This same basic model has the potential to serve as a test bed for implementing any proposed mechanism. Currently, the theory is not complete because the way in which open goals direct the acquisition of information has not been fully specified.

It has been suggested that goals for uncompleted intentions are maintained at a heightened level of activation (Goschke & Kuhl, 1993; Marsh, Hicks, & Bink, 1998). These studies all found that participants responded faster to items that were related to intentions in tasks like a lexical decision task. The argument is that goals spread activation to related items in memory, and that goals which represent intended actions persist as sources of activation even when people move on to other tasks. This kind of mechanism could explain why participants were faster to respond to hints in the lexical decision task even when they had not solved the associated RAT problem. Additionally others have found a similar effect of responding to or naming answers for unsolved RAT problems as compared to neutral words (Beeman & Bowden, 2000; Bowden & Beeman, 1998; Bowden & Jung-Beeman, 2003a; Shames, 1994). However, this mechanism does not explain why hints lead to improvements in both proportion of problems solved and solution time.

One possibility is that open goals set up a pattern in memory through which incoming information is matched against automatically. Items which share some relationship with this pattern have an association with the pattern constructed or strengthened. Therefore when the problem solver returns to the problem there are new associations to explore which may lead to the solution of the problem. The process of recognizing the relevance of a pattern and creating an association to new information could cause the problem solver to become aware of the original problem if there are no other competing task demands. This kind of process could therefore also explain why items related to unsolved problems cause a delay when switching to a new task (Rothermund, 2003).

This process would be selective as it only builds or strengthens associations between the representation in memory and the new information so that it would provide the selectivity in choosing which associations are primed. Kaplan's (1989) theory of incubation involving the diminution of interference and priming would benefit from such a selective priming mechanism. One of the problems with Kaplan's theory is that it does not fully explain how items become primed and incorporated into insight problem solving during incubation.

Priming is generally thought to be a short-term mechanism lasting seconds at most. However, there have been studies which have shown longer term priming effects (Becker et al., 1997; Goschke & Kuhl, 1993; Marsh, Hicks, & Bink, 1998). The two ideas behind these results are the persistence of goals in memory that spread activation that has already been discussed and the idea of semantic attractors in parallel distributed processing networks. The semantic attractor explanation of priming relies on the idea that processing an item places the network in a certain state based on the distribution of activation across units, and a semantically similar item is likely to have a similar representation so the network takes a shorter amount of time to settle to the new state. However, both of these priming effects last on the order of seconds, and any potential mechanism would have to last for at least 5-10 minutes to cover the range of time delays in the studies presented in this dissertation.

The proposed mechanism is that open goals setup a pattern in memory against which incoming information is matched against. The idea of a pattern in memory is similar to the ideas which are present in the gist memory literature (e.g., Brainerd & Reyna, 1993; Brainerd & Reyna, 2004). Essentially the idea is that verbatim and gist memory are different and in some ways independent. People are more likely to show evidence of a false memory when they are prompted with information that is consistent with the gist of what happened while the actual information they are verifying was not actually present in the original experience (Brainerd & Reyna, 2004). Gist memories are more likely to survive over time than verbatim memories as they are apparently less prone to retroactive and proactive interference (Brainerd & Reyna, 2004). The pattern that is setup in memory could be similar to gist memory to the extent that it represents general meanings and associations but not the actual surface structure of the problem. It could then be maintained in memory for longer times just as gist memory is less resistant to interference.

Some kind of combination of long-term priming and a persistent long-term memory structure are necessary to account for the results presented here. The basic idea for an open goal mechanism has been presented, but it is far from a complete and detailed theory. Some potential relationships to other areas of research on cognition have been identified, and they should be informative as the development of an open goal mechanism proceeds. The end theory should be both powerful and parsimonious as it will link problem solving with lower level mechanisms including processes of memory and attention.

### 6.2 Conclusions

The main thesis of this dissertation is that open goals influence problem solving by increasing the chances that problem relevant information will be used by the problem solver. The results of the studies presented here show that having an open goal makes one more likely to make use of information encountered between problem-solving attempts even if that information is not consciously noticed. Processing of problem relevant information is facilitated as it was generally found that participants responded to the hints faster than to neutral words. Making use of presented information was a more reliable indicator of the influence of open goals or unsolved problems than the free recall measure upon which the Zeigarnik effect was based. These findings all support the idea that there is some mechanism by which open goals influence the acquisition and use of problem relevant information.

There are a variety of implications of this work. One is that it should be possible to observe and explain opportunistic behavior in a variety of tasks. As task complexity increases people generally respond by setting up a number of subgoals to deal with parts of the problem. Useful information encountered while working on one of these subgoals should also be incorporated into other subgoals where it would be useful. This situation is one form of transfer. Other forms of transfer including analogical transfer could also be influenced by open goals. Some of the insights that people experience during an incubation period can be explained by the same open goal mechanism. These kinds of insights and analogies could be one source of creative and innovative ideas.

This theory does not imply that all experiences and activities will be useful to solving open problems. The information must somehow be similar enough to the representation of the problem that has been developed during work on the problem. This means that information encountered in activities similar to the problem will probably be more likely to become incorporated into problem solving. For example, a designer may make more progress in overcoming an impasse by looking at a variety of products or working on other design problems, but the designer is less likely to experience insight or make progress by going dancing.

A detailed theory of how open goals influence cognition is itself currently an open goal, but the results presented here have contributed to the beginnings of such a theory. The argument throughout this dissertation has been that the resulting theory will contribute to our understanding of problem solving, memory, and other areas of cognition. Hopefully, continuing to develop this theory will make these contributions a reality and will provide us with insight into areas of cognition that have traditionally been difficult to investigate such as insight and creativity.

# References


Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. L. (2004). An integrated theory of the mind. *Psychological Review, 111*(4), 1036-1060.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum.

Andrews, F. M. (1975). Social and psychological factors which influence the creative process. In I. A. Taylor & J. W. Getzels (Eds.), *Perspectives in creativity* (pp. 117-145). Chicago: Aldine.

Ball, L. J., Evans, J. S. B. T., & Dennis, I. (1994). Cognitive processes in engineering design: A longitudinal study. *Ergonomics, 37*(11), 1753-1786.

Ball, L. J., Evans, J. S. B. T., Dennis, I., & Ormerod, T. C. (1997). Problem-solving strategies and expertise in engineering design. *Thinking & Reasoning, 3*(4), 247-270.

Ball, L. J., & Ormerod, T. C. (1995). Structured and opportunistic processing in design: A critical discussion. *International Journal of Human-Computer Studies, 43*(1), 131-151.

Balota, D. A., Cortese, M. J., Hutchison, K. A., Neely, J. H., Nelson, D., Simpson, G. B., et al. (2002). The English Lexicon Project: A web-based repository of descriptive and behavioral measures for 40,481 English words and nonwords. from http://elexicon.westl.edu/

Becker, S., Moscovitch, M., Behrmann, M., & Joordens, S. (1997). Long-term semantic priming: A computational account and empirical evidence. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 23*(5), 1059-1082.

Beeman, M. J., & Bowden, E. M. (2000). The right hemisphere maintains solution-related activation for yet-to-be-solved problems. *Memory & Cognition, 28*(7), 1231-1241.

Bowden, E. M. (1997). The effect of reportable and unreportable hints on anagram solution and the aha! Experience. *Consciousness and Cognition, 6*(4), 545-573.

Bowden, E. M., & Beeman, M. J. (1998). Getting the right idea: Semantic activation in the right hemisphere may help solve insight problems. *Psychological Science, 9*(6), 435-440.

Bowden, E. M., & Jung-Beeman, M. (2003a). Aha! - Insight experience correlates with solution activation in the right hemisphere. *Psychonomic Bulletin & Review, 10*(3), 730-737.

Bowden, E. M., & Jung-Beeman, M. (2003b). Normative data for 144 compound remote associate problems. *Behavior Research Methods, Instruments, & Computers, 35*(4), 634-639.

Brainerd, C. J., & Reyna, V. F. (1993). Memory independence and memory interference in cognitive development. *Psychological Review, 100*(1), 42-67.

Brainerd, C. J., & Reyna, V. F. (2004). Fuzzy-trace theory and memory development. *Developmental Review, 24*(4), 396-439.

Campbell, D. T. (1960). Blind variation and selective retentions in creative thought as in other knowledge processes. *Psychological Review, 67*, 380-400.

Christensen, B. T., & Schunn, C. D. (2005). Spontaneous access and analogical incubation effects. *Creativity Research Journal, 17*(2-3), 207-220.

Connor, L. T., Balota, D. A., & Neely, J. H. (1992). On the relation between feeling of knowing and lexical decision: Persistent subthreshold activation or topic familiarity? *Journal of Experimental Psychology: Learning, Memory, & Cognition, 18*(3), 544-554.

Craik, F. I., & Lockhart, R. S. (1972). Levels of processing: A framework for memory research. *Journal of Verbal Learning & Verbal Behavior, 11*(6), 671-684.

Daily, L. Z., Lovett, M. C., & Reder, L. M. (2001). Modeling individual differences in working memory performance: A source activation account. *Cognitive Science, 25*(3), 315-353.

Dodds, R. A., Smith, S. M., & Ward, T. B. (2002). The use of environmental clues during incubation. *Creativity Research Journal, 14*(3-4), 287-304.

Dodds, R. A., Ward, T. B., & Smith, S. M. (in press). A review of the experimental literature on incubation in problem solving and creativity. In M. A. Runco (Ed.), *Creativity research handbook* (Vol. 3). Cresskill, NJ: Hampton Press.

Dominowski, R. L., & Jenrick, R. (1972). Effects of hints and interpolated activity on solution of an insight problem. *Psychonomic Science, Vol. 26*(6), 335-338.

Dorfman, J., Shames, V. A., & Kihlstrom, J. F. (1996). Intuition, incubation, and insight: Implicit cognition in problem solving. In G. D. M. Underwood (Ed.), *Implicit cognition* (pp. 257-296). London: London University Press.

Dreistadt, R. (1969). The use of analogies and incubation in obtaining insights in creative problem solving. *Journal of Psychology: Interdisciplinary and Applied, 71*(2), 159-175.

Dunbar, K. (2001). The analogical paradox: Why analogy is so easy in naturalistic settings yet so difficult in the psychological laboratory. In D. Gentner, K. J. Holyoak & B. N. Kokinov (Eds.), *The analogical mind: Perspectives from cognitive science* (pp. 313-334). Cambridge, MA: MIT Press.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (rev. ed.). Cambridge, MA: The MIT Press.

Fellbaum, C. (Ed.). (1998). *WordNet: An electronic lexical database*. Cambridge, MA: MIT Press.

Forbus, K. D., Gentner, D., & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science, 19*(2), 141-205.

Fulgosi, A., & Guilford, J. P. (1968). Short-term incubation in divergent production. *American Journal of Psychology, 81*(2), 241-246.

Gick, M. L., & Holyoak, K. J. (1980). Analogical problem solving. *Cognitive Psychology, 12*(3), 306-355.

Gick, M. L., & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology, 15*(1), 1-38.

Goschke, T., & Kuhl, J. (1993). Representation of intentions: Persisting activation in memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 19*(5), 1211-1226.

Guindon, R. (1990). Designing the design process: Exploiting opportunistic thoughts. *Human-Computer Interaction, 5*, 305-344.

Holyoak, K. J., & Koh, K. (1987). Surface and structural similarity in analogical transfer. *Memory & Cognition, 15*(4), 332-340.

Kane, M. J., Hambrick, D. Z., Tuholski, S. W., Wilhelm, O., Payne, T. W., & Engle, R. W. (2004). The generality of working memory capacity: A latent-variable approach to verbal and visuospatial memory span and reasoning. *Journal of Experimental Psychology-General, 133*(2), 189-217.

Kaplan, C. A. (1989). *Hatching a theory of incubation: Does putting a problem aside really help? If so, why?* Unpublished Ph.D., Carnegie Mellon University, Pittsburgh, PA.

Kaplan, C. A., & Simon, H. A. (1990). In search of insight. *Cognitive Psychology, 22*(3), 374-419.

Kershaw, T. C., & Ohlsson, S. (2004). Multiple causes of difficulty in insight: The case of the nine-dot problem. *Journal of Experimental Psychology-Learning Memory and Cognition, 30*(1), 3-13.

Kihlstrom, J. F., Shames, V. A., & Dorfman, J. (1996). Intimations of memory and thought. In L. M. Reder (Ed.), *Implicit memory and metacognition* (pp. 1-23). Mahwah, NJ: Erlbaum.

Klinger, E. (1996). The contents of thoughts: Interference as the downside of adaptive normal mechanisms in thought flow. In I. G. Sarason, G. R. Pierce & B. R. Sarason (Eds.), *Cognitive interference: Theories, methods, and findings* (pp. 3-23). Hillsdale, NJ: Lawrence Erlbaum Associates.

Knoblich, G., Ohlsson, S., Haider, H., & Rhenius, D. (1999). Constraint relaxation and chunk decomposition in insight problem solving. *Journal of Experimental Psychology-Learning Memory and Cognition, 25*(6), 1534-1555.

Kotovsky, K. (2003). Problem solving - Large/small, hard/easy, conscious/nonconscious, problem-space/problem-solver: Issues of dichotomization. In J. E. Davidson & R. J. Sternberg (Eds.), *The psychology of problem solving* (pp. 373-383). New York, NY: Cambridge University Press.

Langley, P., & Jones, R. (1988). A computational model of scientific insight. In R. J. Sternberg (Ed.), *The nature of creativity: Contemporary psychological perspectives* (pp. 177-201). New York, NY, US: Cambridge University Press.

Lockhart, R. S., Lamon, M., & Gick, M. L. (1988). Conceptual transfer in simple insight problems. *Memory & Cognition, 16*(1), 36-44.

Lovett, M. C., Daily, L. Z., & Reder, L. M. (2001). A source activation theory of working memory: Cross-task prediction of performance in ACT-R. *Cognitive Systems Research, 1*(1-4), 99-118.

Lovett, M. C., Reder, L. M., & Lebiere, C. (1999). Modeling working memory in a unified architecture: An ACT-R perspective. In A. Miyake & P. Shah (Eds.), *Models of working memory: Mechanisms of active maintenance and executive control* (pp. 135-182). New York, NY, US: Cambridge University Press.

Lovett, M. C., & Schunn, C. D. (1999). Task representations, strategy variability, and base-rate neglect. *Journal of Experimental Psychology-General, 128*(2), 107-130.

MacGregor, J. N., Ormerod, T. C., & Chronicle, E. P. (2001). Information processing and insight: A process model of performance on the nine-dot and related problems. *Journal of Experimental Psychology-Learning Memory and Cognition, 27*(1), 176-201.

Maier, N. R. F. (1931). Reasoning in humans II: The solution of a problem and its appearance in consciousness. *Journal of Comparative Psychology, 12*, 181-194.

Marsh, R. L., Hicks, J. L., & Bink, M. L. (1998). Activation of completed, uncompleted, and partially completed intentions. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 24*(2), 350-361.

Marsh, R. L., Hicks, J. L., & Brink, M. L. (1998). The activation of completed, uncompleted, and partially completed intentions. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 24*, 350-361.

Mednick, M. T., Mednick, S. A., & Mednick, E. V. (1964). Incubation of creative performance and specific associative priming. *Journal of Abnormal & Social Psychology, 69*(1), 84-88.

Mednick, S. A. (1962). The associative basis of the creative process. *Psychological Review, 69*, 220-232.

Metcalfe, J. (1986a). Feeling of knowing in memory and problem solving. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12*(2), 288-294.

Metcalfe, J. (1986b). Premonitions of insight predict impending error. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 12*(4), 623-634.

Moss, J., Kotovsky, K., & Cagan, J. (2006). The role of functionality in the mental representations of engineering students: Some differences in the early stages of expertise. *Cognitive Science, 30*(1), 65-93.

Nelson, D. L., McEvoy, C. L., & Schreiber, T. A. (1998). The University of South Florida word association, rhyme, and word fragment norms. from http://www.usf.edu/FreeAssociation

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Oxford: Prentice-Hall.

Novick, L. R. (1988). Analogical transfer, problem similarity, and expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 14*(3), 510-520.

Ohlsson, S. (1992). Information-processing explanations of insight and related phenomena. In M. T. Keane & K. J. Gilhooly (Eds.), *Advances in the psychology of thinking* (pp. 1-43). London: Harvester Wheatsheaf.

Olton, R. M. (1979). Experimental studies of incubation: Searching for the elusive. *Journal of Creative Behavior, 13*(1), 9-22.

Olton, R. M., & Johnson, D. M. (1976). Mechanisms of incubation in creative problem solving. *American Journal of Psychology, 89*(4), 617-630.

Ormerod, T. C., MacGregor, J. N., & Chronicle, E. P. (2002). Dynamics and constraints in insight problem solving. *Journal of Experimental Psychology-Learning Memory and Cognition, 28*(4), 791-799.

Patalano, A. L., & Seifert, C. M. (1994). Memory for impasses during problem solving. *Memory & Cognition, 22*(2), 234-242.

Patalano, A. L., & Seifert, C. M. (1997). Opportunistic planning: Being reminded of pending goals. *Cognitive Psychology, 34*(1), 1-36.

Plaut, D. C., & Booth, J. R. (2000). Individual and developmental differences in semantic priming: Empirical and computational support for a single-mechanism account of lexical processing. *Psychological Review, 107*(4), 786-823.

Rothermund, K. (2003). Automatic vigilance for task-related information: Perseverance after failure and inhibition after success. *Memory & Cognition, 31*(3), 343-352.

Schneider, W., Eschman, A., & Zuccolotto, A. (2002). *E-Prime user's guide*. Pittsburgh: Psychology Software Tools Inc.

Schooler, J. W., Ohlsson, S., & Brooks, K. (1993). Thoughts beyond words: When language overshadows insight. *Journal of Experimental Psychology: General, 122*(2), 166-183.

Schunn, C. D., & Dunbar, K. (1996). Priming, analogy, and awareness in complex reasoning. *Memory & Cognition, 24*(3), 271-284.

Seifert, C. M., Meyer, D. E., Davidson, N., Patalano, A. L., & Yaniv, I. (1995). Demystification of cognitive insight: Opportunistic assimilation and the prepared-mind perspective. In R. J. Sternberg & J. E. Davidson (Eds.), *The nature of insight* (pp. 65-124). Cambridge, MA: MIT Press.

Seifert, C. M., & Patalano, A. L. (1991). Memory for incomplete tasks: A re-examination of the Zeigarnik effect, *Thirteenth Annual Conference of the Cognitive Science Society* (pp. 114-119). Hillsdale, NJ: Erlbaum.

Shah, J. Y., & Kruglanski, A. W. (2002). Priming against your will: How accessible alternatives affect goal pursuit. *Journal of Experimental Social Psychology, 38*, 368-383.

Shames, V. A. (1994). *Is there such a thing as implicit problem-solving?* Unpublished doctoral dissertation, University of Arizona, Tucson.

Simon, H. A. (1966). Scientific discovery and the psychology of problem solving. In R. G. Colodny (Ed.), *Mind and cosmos: Essays in contemporary science and philosophy* (pp. 22-40). Pittsburgh, PA: University of Pittsburgh Press.

Simon, H. A. (1969). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Simon, H. A. (1973). Structure of ill structured problems. *Artificial Intelligence, 4*(3-4), 181-201.

Simonton, D. K. (1999). Creativity as blind variation and selective retention: Is the creative process Darwinian? *Psychological Inquiry, 10*(4), 309-328.

Smith, S. M. (1995a). Fixation, incubation, and insight in memory and creative thinking. In S. M. Smith, T. B. Ward & R. A. Finke (Eds.), *The creative cognition approach*. Cambridge, MA, US: The MIT Press.

Smith, S. M. (1995b). Getting into and out of mental ruts: A theory of fixation, incubation, and insight. In R. J. Sternberg & J. E. Davidson (Eds.), *The nature of insight*. Cambridge, MA: MIT Press.

Smith, S. M., & Blankenship, S. E. (1991). Incubation and the persistence of fixation in problem solving. *American Journal of Psychology, 104*(1), 61-87.

Sternberg, R. J., & Davidson, J. E. (Eds.). (1995). *The nature of insight*. Cambridge, MA, US: The MIT Press.

Van Bergen, A. (1968). *Task interruption*. Amsterdam: North-Holland.

Visser, W. (1990). More-or-less following a plan during design: Opportunistic deviations in specification. *International Journal of Man-Machine Studies, 33*(3), 247-278.

Visser, W. (1996). Two functions of analogical reasoning in design: A cognitive psychology approach. *Design Studies, 17*, 417-434.

Wallas, G. (1926). *The art of thought*. London: J. Cape.

Weisberg, R. W. (1988). Problem solving and creativity. In R. J. Sternberg (Ed.), *The nature of creativity: Contemporary psychological perspectives* (pp. 148-176). New York, NY: Cambridge University Press.

Weisberg, R. W. (1995). Prolegomena to theories of insight in problem solving: A taxonomy of problems. In R. J. Sternberg & J. E. Davidson (Eds.), *The nature of insight* (pp. 157-196). Cambridge, MA: MIT Press.

Wiley, J. (1998). Expertise as mental set: The effects of domain knowledge in creative problem solving. *Memory & Cognition, 26*(4), 716-730.

Yaniv, I., & Meyer, D. E. (1987). Activation and metacognition of inaccessible stored information: Potential bases for incubation effects in problem solving. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 13*(2), 187-205.

Yaniv, I., Meyer, D. E., & Davidson, N. S. (1995). Dynamic memory processes in retrieving answers to questions: Recall failures, judgments of knowing, and acquisition of information. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 21*(6), 1509-1521.

Zeigarnik, B. (1938). On finished and unfinished tasks. In W. D. Ellis (Ed.), *A source book of gestalt psychology* (pp. 300-314). New York: Harcourt-Brace. (Reprinted and condensed from Psychologische Forschung, 9, 1-85, by B. Zeigarnik, 1927)

# Appendix I: RAT Problems Used In Studies

**Study 1A**

| Word 1 | Word 2 | Word 3 | Answer |
|---|---|---|---|
| change | circuit | cake | short |
| wet | law | business | suit |
| dust | cereal | fish | bowl |
| iron | shovel | engine | steam |
| hold | print | stool | foot |
| fox | man | peep | hole |
| right | cat | carbon | copy |
| main | sweeper | light | street |
| pike | coat | signal | turn |
| time | blown | nelson | full |
| hammer | gear | hunter | head |
| type | ghost | screen | writer |
| tomato | bomb | picker | cherry |
| wagon | break | radio | station |
| health | taker | less | care |
| marshal | children | piano | grand |
| eight | skate | stick | figure |
| down | question | check | mark |
| animal | back | rat | pack |
| pile | market | room | stock |

**Study 1B**

| Word 1 | Word 2 | Word 3 | Answer |
|---|---|---|---|
| fox | man | peep | hole |
| main | sweeper | light | street |
| pike | coat | signal | turn |
| way | board | sleep | walk |
| opera | hand | dish | soap |
| eight | skate | stick | figure |
| wet | law | business | suit |
| hammer | gear | hunter | head |
| horse | human | drag | race |
| boot | summer | ground | camp |
| type | ghost | screen | writer |
| mill | tooth | dust | saw |
| wagon | break | radio | station |
| blank | list | mate | check |
| animal | back | rat | pack |

| | | | |
|---|---|---|---|
| house | thumb | pepper | green |
| iron | shovel | engine | steam |
| piece | mind | dating | game |
| right | cat | carbon | copy |
| tomato | bomb | picker | cherry |
| health | taker | less | care |
| officer | cash | larceny | petty |
| time | blown | nelson | full |
| pile | market | room | stock |
| teeth | arrest | start | false |
| change | circuit | cake | short |
| tank | hill | secret | top |
| marshal | child | piano | grand |
| cover | arm | wear | under |
| sense | courtesy | place | common |

**Study 2**

| **Word 1** | **Word 2** | **Word 3** | **Answer** |
|---|---|---|---|
| fox | man | peep | hole |
| main | sweeper | light | street |
| pike | coat | signal | turn |
| way | board | sleep | walk |
| opera | hand | dish | soap |
| eight | skate | stick | figure |
| wet | law | business | suit |
| hammer | gear | hunter | head |
| horse | human | drag | race |
| boot | summer | ground | camp |
| type | ghost | screen | writer |
| mill | tooth | dust | saw |
| wagon | break | radio | station |
| blank | list | mate | check |
| animal | back | rat | pack |
| house | thumb | pepper | green |
| iron | shovel | engine | steam |
| piece | mind | dating | game |
| right | cat | carbon | copy |
| tomato | bomb | picker | cherry |
| health | taker | less | care |
| officer | cash | larceny | petty |
| time | blown | nelson | full |
| pile | market | room | stock |
| change | circuit | cake | short |

| | | | |
|---|---|---|---|
| tank | hill | secret | top |
| marshal | child | piano | grand |
| cover | arm | wear | under |
| sense | courtesy | place | common |

**Study 3**

| **Word 1** | **Word 2** | **Word 3** | **Answer** |
|---|---|---|---|
| fox | man | peep | hole |
| main | sweeper | light | street |
| pike | coat | signal | turn |
| way | board | sleep | walk |
| opera | hand | dish | soap |
| eight | skate | stick | figure |
| wet | law | business | suit |
| hammer | gear | hunter | head |
| horse | human | drag | race |
| boot | summer | ground | camp |
| type | ghost | screen | writer |
| mill | tooth | dust | saw |
| wagon | break | radio | station |
| blank | list | mate | check |
| animal | back | rat | pack |
| house | thumb | pepper | green |
| iron | shovel | engine | steam |
| piece | mind | dating | game |
| right | cat | carbon | copy |
| tomato | bomb | picker | cherry |
| health | taker | less | care |
| officer | cash | larceny | petty |
| time | blown | nelson | full |
| pile | market | room | stock |
| teeth | arrest | start | false |
| change | circuit | cake | short |
| tank | hill | secret | top |
| marshal | child | piano | grand |
| cover | arm | wear | under |
| sense | courtesy | place | common |

# Appendix II: Model Code

```
(setf *tenured-bytes-limit* 100000000)
(defconstant *duration* 30)
(setf *correct* 0)
(setf *answers* nil)
(setf *time* 0)
(setf *problem-words* nil)
(setf *other-words* nil)
(setf *problems* nil)
(setf *incorrect-problems* nil)
(setf *filler-words* nil)
(setf *nonwords* nil)
(setf *results-list* nil)
(setf *retrieval-results-list* nil)
 (setf *problem-fans* '((17 105 4) (18 4 68) (2 29 14) (51 76 18) (10 87 17) (3 9 34) (6 36 34) (7 26 9) (41 9 7) (15 7
39) (12 5 18) (19 24 18) (14 32 20) (5 17 17) (34 153 16) (87 7 21) (15 8 149) (12 32 2) (11 1 23) (34 30 8) (2 5 12)
(121 4 3) (11 10 4) (5 29 17) (21 21 12) (16 15 22) (30 6 20) (9 31 85) (13 13 11) (20 24 23)))

(defun build-initial-display (word1 word2 word3)
  (setf *experiment-window* (open-exp-window "RAT"
                                             :visible nil
                                             :width 300
                                             :height 300))
  (add-text-to-exp-window :x 140 :y 100 :text (string word1))
  (add-text-to-exp-window :x 140 :y 130 :text (string word2))
  (add-text-to-exp-window :x 140 :y 160 :text (string word3))
  (install-device *experiment-window*)
  (proc-display))

(defun build-display (word1 word2 word3)
  (clear-exp-window)
  (add-text-to-exp-window :x 140 :y 100 :text (string word1))
  (add-text-to-exp-window :x 140 :y 130 :text (string word2))
  (add-text-to-exp-window :x 140 :y 160 :text (string word3))
  (proc-display))

(defun setup-dm ()
  (setf *problems* nil)
  (setf *problems2* nil)
  (setf *problem-words* nil)
  (setf *other-words* nil)
  (setf *filler-words* nil)
  (setf *nonwords* nil)
  (setf *unseen-problems* nil)
  (add-problem-words (* 3 (length *problem-fans*)))
  (add-other-words 500)
  (dolist (i *problem-fans*)
    (add-problem (car i) (cadr i) (caddr i)))
  (setf *problems* (permute-list *problems*))
  (dotimes (i 10)
    (push (pop *problems*) *unseen-problems*))
  (setup-blocks)
```

```
)

(defun setup-blocks ()
  (let* ((extra-unseen (1+ (act-r-random 4)))
         (extra-unseen2 (cond ((= extra-unseen 1)
                               (+ 2 (act-r-random 3)))
                              ((= extra-unseen 2)
                               (let ((temp (act-r-random 3)))
                                 (cond ((= temp 0) 1)
                                       (t (+ temp 2)))))
                              ((= extra-unseen 3)
                               (let ((temp (act-r-random 3)))
                                 (cond ((= temp 2) 4)
                                       (t (+ temp 1)))))
                              ((= extra-unseen 4)
                               (+ 1 (act-r-random 3)))
                              (t
                               (print "shouldn't be here in setup-blocks")))))
    (setf *block1* (subseq *problems* 0 5))
    (setf *block2* (subseq *problems* 5 10))
    (setf *block3* (subseq *problems* 10 15))
    (setf *block4* (subseq *problems* 15 20))
    (setf *block12* (permute-list (append *block1* (subseq *unseen-problems* 0 2))))
    (setf *block22* (permute-list (append *block2* (subseq *unseen-problems* 2 4))))
    (setf *block32* (permute-list (append *block3* (subseq *unseen-problems* 4 6))))
    (setf *block42* (permute-list (append *block4* (subseq *unseen-problems* 6 8))))
    (setf *hints12* (list (car (nth 1 *unseen-problems*))))
    (setf *hints22* (list (car (nth 3 *unseen-problems*))))
    (setf *hints32* (list (car (nth 5 *unseen-problems*))))
    (setf *hints42* (list (car (nth 7 *unseen-problems*))))
    (setf *notHints12* (list (car (nth 0 *unseen-problems*))))
    (setf *notHints22* (list (car (nth 2 *unseen-problems*))))
    (setf *notHints32* (list (car (nth 4 *unseen-problems*))))
    (setf *notHints42* (list (car (nth 6 *unseen-problems*))))
    (cond ((= extra-unseen 1)
           (push (nth 8 *unseen-problems*) *block12*)
           (push (car (nth 8 *unseen-problems*)) *notHints12*))
          ((= extra-unseen 2)
           (push (nth 8 *unseen-problems*) *block22*)
           (push (car (nth 8 *unseen-problems*)) *notHints22*))
          ((= extra-unseen 3)
           (push (nth 8 *unseen-problems*) *block32*)
           (push (car (nth 8 *unseen-problems*)) *notHints32*))
          ((= extra-unseen 4)
           (push (nth 8 *unseen-problems*) *block42*)
           (push (car (nth 8 *unseen-problems*)) *notHints42*)))
    (cond ((= extra-unseen2 1)
           (push (nth 9 *unseen-problems*) *block12*)
           (push (car (nth 9 *unseen-problems*)) *hints12*))
          ((= extra-unseen2 2)
           (push (nth 9 *unseen-problems*) *block22*)
           (push (car (nth 9 *unseen-problems*)) *hints22*))
          ((= extra-unseen2 3)
           (push (nth 9 *unseen-problems*) *block32*)
```

```
          (push (car (nth 9 *unseen-problems*)) *hints32*))
         ((= extra-unseen2 4)
          (push (nth 9 *unseen-problems*) *block42*)
          (push (car (nth 9 *unseen-problems*)) *hints42*))))
  (setf *current-block* 1)
  (setf *problems* *block1*)
  (setf *hints* *hints12*)
  (setf *notHints* *notHints12*))

(defun advance-block ()
  (cond ((= *current-block* 1)
         (setf *task* "RAT2")
         (setf *problems* (permute-list *block12*))
         (setf *hints* nil)
         (setf *notHints* nil)
         (setf *current-block* 12))
        ((= *current-block* 2)
         (setf *task* "RAT2")
         (setf *problems* (permute-list *block22*))
         (setf *hints* nil)
         (setf *notHints* nil)
         (setf *current-block* 22))
        ((= *current-block* 3)
         (setf *task* "RAT2")
         (setf *problems* (permute-list *block32*))
         (setf *hints* nil)
         (setf *notHints* nil)
         (setf *current-block* 32))
        ((= *current-block* 4)
         (setf *task* "RAT2")
         (setf *problems* (permute-list *block42*))
         (setf *hints* nil)
         (setf *notHints* nil)
         (setf *current-block* 42))
        ((= *current-block* 12)
         (setf *task* "RAT")
         (setf *problems* (permute-list *block2*))
         (setf *hints* *hints22*)
         (setf *notHints* *notHints22*)
         (setf *current-block* 2))
        ((= *current-block* 22)
         (setf *task* "RAT")
         (setf *problems* (permute-list *block3*))
         (setf *hints* *hints32*)
         (setf *notHints* *notHints32*)
         (setf *current-block* 3))
        ((= *current-block* 32)
         (setf *task* "RAT")
         (setf *problems* (permute-list *block4*))
         (setf *hints* *hints42*)
         (setf *notHints* *notHints42*)
         (setf *current-block* 4))
        ((= *current-block* 42)
         (print "shouldn't be here in advance-block")
```

```
              (break))))

(defun add-problem-words (n)
  (dotimes (i n)
    (let ((new-word (gentemp 'word))
              (new-word-mapping (gentemp 'word-mapping))
              (new-meaning (gentemp 'meaning)))
      (add-dm-fct (list (list new-meaning 'isa 'meaning)))
      (add-dm-fct (list (list new-word 'isa 'word 'meaning1 new-meaning 'self-ref new-word)))
      (add-dm-fct (list (list new-word-mapping 'isa 'word-mapping
                                        'word new-word
                                        'text (string new-word))))
      (set-base-levels-fct (list (list new-word-mapping 5000 -10000)))
      (set-base-levels-fct (list (list new-word 125 -100000)))
      (push new-word *problem-words*)
      (dotimes (i (1+ (act-r-random 15)))
             (let ((new-word2 (gentemp'word))
                 (new-word-mapping2 (gentemp 'word-mapping)))
               (add-dm-fct (list (list new-word2 'isa 'word 'meaning1 new-meaning 'self-ref new-word2)))
               (add-dm-fct (list (list new-word-mapping2 'isa 'word-mapping
                                        'word new-word2
                                        'text (string new-word2))))
               (set-base-levels-fct (list (list new-word-mapping2 5000 -10000)))
               (set-base-levels-fct (list (list new-word2 125 -100000)))
               (push new-word2 *other-words*))))))

(defun add-other-words (n)
  (dotimes (i n)
    (let ((new-word (gentemp 'word))
              (new-word-mapping (gentemp 'word-mapping))
              (new-meaning (gentemp 'meaning)))
      (add-dm-fct (list (list new-meaning 'isa 'meaning)))
      (add-dm-fct (list (list new-word 'isa 'word 'meaning1 new-meaning 'self-ref new-word)))
      (add-dm-fct (list (list new-word-mapping 'isa 'word-mapping
                                        'word new-word
                                        'text (string new-word))))
      (set-base-levels-fct (list (list new-word-mapping 5000 -10000)))
      (set-base-levels-fct (list (list new-word 125 -100000)))
      (push new-word *other-words*)))
  (dotimes (i 100)
    (let ((new-word (gentemp 'word))
              (new-word-mapping (gentemp 'word-mapping))
              (new-meaning (gentemp 'meaning)))
      (add-dm-fct (list (list new-meaning 'isa 'meaning)))
      (add-dm-fct (list (list new-word 'isa 'word 'meaning1 new-meaning 'self-ref new-word)))
      (add-dm-fct (list (list new-word-mapping 'isa 'word-mapping
                                        'word new-word
                                        'text (string new-word))))
      (set-base-levels-fct (list (list new-word-mapping 5000 -10000)))
      (set-base-levels-fct (list (list new-word 125 -100000)))
      (push (string new-word) *filler-words*)))
  (dotimes (i 100)
    (let ((new-word (gentemp 'word)))
      (push (string new-word) *nonwords*))))
```

```
(defun add-problem (x y z)
  (print "add-problem")
  (let* ((words (permute-list *problem-words*))
             (answer (gentemp 'word))
             (new-word-mapping (gentemp 'word-mapping))
             (others (permute-list *other-words*)))
    (setf *problem-words* (cdddr words))
    (let ((new-meaning1 (chunk-slot-value-fct (car words) 'meaning1))
              (new-meaning2 (chunk-slot-value-fct (cadr words) 'meaning1))
              (new-meaning3 (chunk-slot-value-fct (caddr words) 'meaning1)))
      (add-dm-fct (list (list answer 'isa 'word 'meaning1 new-meaning1
                                    'meaning2 new-meaning2 'meaning3 new-meaning3
                                    'self-ref answer 'problem-ref (string (car words)))))
      (add-dm-fct (list (list new-word-mapping 'isa 'word-mapping
                                    'word answer
                                    'text (string answer))))
      (set-base-levels-fct (list (list new-word-mapping 5000 -10000)))
      (set-base-levels-fct (list (list answer 125 -100000)))
      (dotimes (i x)
          (let ((new-word2 (gentemp'word))
               (new-word-mapping2 (gentemp 'word-mapping)))
            (add-dm-fct (list (list new-word2 'isa 'word 'meaning1 new-meaning1
                                        'self-ref new-word2 'problem-ref (string (car words)))))
            (add-dm-fct (list (list new-word-mapping2 'isa 'word-mapping
                                        'word new-word2
                                        'text (string new-word2))))
            (set-base-levels-fct (list (list new-word-mapping2 5000 -10000)))
            (set-base-levels-fct (list (list new-word2 125 -100000)))
            (push new-word2 *other-words*)))
      (dotimes (i y)
          (let ((new-word2 (gentemp'word))
               (new-word-mapping2 (gentemp 'word-mapping)))
            (add-dm-fct (list (list new-word2 'isa 'word 'meaning1 new-meaning2
                                        'self-ref new-word2 'problem-ref (string (car words)))))
            (add-dm-fct (list (list new-word-mapping2 'isa 'word-mapping
                                        'word new-word2
                                        'text (string new-word2))))
            (set-base-levels-fct (list (list new-word-mapping2 5000 -10000)))
            (set-base-levels-fct (list (list new-word2 125 -100000)))
            (push new-word2 *other-words*)))
      (dotimes (i z)
          (let ((new-word2 (gentemp'word))
               (new-word-mapping2 (gentemp 'word-mapping)))
            (add-dm-fct (list (list new-word2 'isa 'word 'meaning1 new-meaning3
                                        'self-ref new-word2 'problem-ref (string (car words)))))
            (add-dm-fct (list (list new-word-mapping2 'isa 'word-mapping
                                        'word new-word2
                                        'text (string new-word2))))
            (set-base-levels-fct (list (list new-word-mapping2 5000 -10000)))
            (set-base-levels-fct (list (list new-word2 125 -100000)))
            (push new-word2 *other-words*))))
    (setf others (permute-list *other-words*))
    (add-compounds (car words) answer x others)
```

```
  (add-compounds (cadr words) answer y others)
  (add-compounds (caddr words) answer z (nthcdr (+ x y) others))
  (push (list answer (car words) (cadr words) (caddr words) (string (car words)))
        *problems*)
  (push (list answer (car words) (cadr words) (caddr words) (string (car words)))
        *problems2*)))

(defun add-compounds (word answer n others)
 (let ((random-list others)
       (new-cwa (gentemp 'compound-word)))
  (add-dm-fct (list (list new-cwa 'isa 'compound-word
                         'comp1 word 'comp2 answer
                         'value (concatenate 'string (string word) (string answer))
                         'meaning (chunk-slot-value-fct word 'meaning1))))
  (set-base-levels-fct (list (list new-cwa 100 -100000)))
  (dotimes (i (- n 1))
   (let ((new-cw (gentemp 'compound-word)))
     (add-dm-fct (list (list new-cw 'isa 'compound-word
                            'comp1 word 'comp2 (nth i random-list)
                            'value (concatenate 'string (string word)
                                                (string (nth i random-list)))
                            'meaning (chunk-slot-value-fct word 'meaning1))))
     (set-base-levels-fct (list (list new-cw 100 -100000)))))))

(defun do-set (m)
 (setf *results-list* nil)
 (setf *retrieval-results-list* nil)
 (dotimes (i m)
  (print i)
  (do-problems)
  (print-results)
  (average-retrieval-records))
 (average-results m))

(defun reset-variables ()
 (setf *retrieval-results-list* nil)
 (setf *numberHints* 0)
 (setf *number-unsolved-trials* 0)
 (setf *number-word-trials* 0)
 (setf *number-nonword-trials* 0)
 (setf *correct-unsolved-trials* 0)
 (setf *correct-word-trials* 0)
 (setf *correct-nonword-trials* 0)
 (setf *number-unseen-trials* 0)
 (setf *correct-unseen-trials* 0)
 (setf *unsolved-problems* nil)
 (setf *extra-hint* nil)
 (setf *correct* 0)
 (setf *correct-sum* 0)
 (setf *already-solved* 0)
 (setf *already-solved-sum* 0)
 (setf *improvednh-sum* 0)
 (setf *improvedh-sum* 0)
 (setf *solvedh-sum* 0)
```

```
  (setf *solvednh-sum* 0)
  (setf *nonword-rt* 0)
  (setf *word-rt* 0)
  (setf *unsolved-rt* 0)
  (setf *unseen-rt* 0)
  (setf *unsolved* 0)
  (setf *improvednh* 0)
  (setf *improvedh* 0)
  (setf *solvedh* 0)
  (setf *solvednh* 0)
  (setf *answers* nil)
  (setf *solvedp-alist* nil)
  (setf *hintp-alist* nil)
  (setf *initial-retrievals* nil)
  (setf *unsolvednh-retrievals* nil)
  (setf *unsolvedh-retrievals* nil)
  (setf *unseennh-retrievals* nil)
  (setf *unseenh-retrievals* nil)
  (setf *solved-retrievals* nil)
  (setf *retrieved-chunks* nil)
  (setf *retrieved-chunks-list* nil))

(defun do-problems ()
  (reset)
  (setup-dm)
  (setf *numberHints* 0)
  (setf *number-unsolved-trials* 0)
  (setf *number-word-trials* 0)
  (setf *number-nonword-trials* 0)
  (setf *correct-unsolved-trials* 0)
  (setf *correct-word-trials* 0)
  (setf *correct-nonword-trials* 0)
  (setf *number-unseen-trials* 0)
  (setf *correct-unseen-trials* 0)
  (setf *unsolved-problems* nil)
  (setf *extra-hint* nil)
  (setf *correct* 0)
  (setf *correct-sum* 0)
  (setf *already-solved* 0)
  (setf *already-solved-sum* 0)
  (setf *improvednh-sum* 0)
  (setf *improvedh-sum* 0)
  (setf *solvedh-sum* 0)
  (setf *solvednh-sum* 0)
  (setf *nonword-rt* 0)
  (setf *word-rt* 0)
  (setf *unsolved-rt* 0)
  (setf *unseen-rt* 0)
  (setf *unsolved* 0)
  (setf *improvednh* 0)
  (setf *improvedh* 0)
  (setf *solvedh* 0)
  (setf *solvednh* 0)
  (setf *answers* nil)
```

```
 (setf *problems* (permute-list *problems*))
 (setf *solvedp-alist* nil)
 (setf *hintp-alist* nil)
 (setf *initial-retrievals* nil)
 (setf *unsolvednh-retrievals* nil)
 (setf *unsolvedh-retrievals* nil)
 (setf *unseennh-retrievals* nil)
 (setf *unseenh-retrievals* nil)
 (setf *solved-retrievals* nil)
 (setf *retrieved-chunks* nil)
 (setf *retrieved-chunks-list* nil)
 (display-first-problem))

(defmethod rpm-window-key-event-handler ((win rpm-window) key)
 (setf *response* (string key))
 (cond ((equal *task* "RAT")
        (cond ((equal *response* (string #\Newline))
               (cond ((equal *answer* (car *current-problem*))
                      (setf *time-elapsed* (- (mp-time) *start-time*))
                      (cond ((< *time-elapsed* 30)
                             (delete-event *end-event*)
                             (push (cons (car *current-problem*) 1) *solvedp-alist*)
                             (setf *response-time* (mp-time))
                             (setf *correct* (1+ *correct*))
                             (setf *correct-sum* (+ *correct-sum* *time-elapsed*))
                             (process-retrieval-record)
                             (display-feedback "solved"))
                            (t
                             (print "answered late"))))
                     (t
                      (print "incorrect answer")
                      (break)
                      (new-tone-sound 500 .5))))))
       ((equal *task* "RAT2")
        (cond ((equal *response* (string #\Newline))
               (cond ((equal *answer* (car *current-problem*))
                      (setf *time-elapsed* (- (mp-time) *start-time*))
                      (cond ((< *time-elapsed* 30)
                             (delete-event *end-event*)
                             (cond ((equal (cdr (assoc *answer* *solvedp-alist*)) 0)
                                    (cond ((equal (cdr (assoc *answer* *hintp-alist*)) 0)
                                           (setf *improvednh* (1+ *improvednh*))
                                           (setf *improvednh-sum* (+ *time-elapsed* *improvednh-sum*))
                                           (print "improved no-hint"))
                                          ((equal (cdr (assoc *answer* *hintp-alist*)) 1)
                                           (setf *improvedh* (1+ *improvedh*))
                                           (setf *improvedh-sum* (+ *time-elapsed* *improvedh-sum*))
                                           (print "improved hint"))
                                          (t
                                           (print "shouldn't be here1\n"))))
                                   ((equal (cdr (assoc *answer* *solvedp-alist*)) 1)
                                    (print "already solved")
                                    (incf *already-solved*)
                                    (setf *already-solved-sum* (+ *time-elapsed* *already-solved-sum*)))
```

```
                                        (t
                                         (cond ((equal (cdr (assoc *answer* *hintp-alist*)) -1)
                                                (setf *solvedh* (1+ *solvedh*))
                                                (setf *solvedh-sum* (+ *time-elapsed* *solvedh-sum*))
                                                (print "solved hint"))
                                               ((equal (cdr (assoc *answer* *hintp-alist*)) -2)
                                                (setf *solvednh* (1+ *solvednh*))
                                                (setf *solvednh-sum* (+ *time-elapsed* *solvednh-sum*))
                                                (print "solved no hint"))
                                               (t
                                                (print "shouldn't be here2\n")))))
                                  (process-retrieval-record)
                                  (display-feedback "solved"))
                                 (t
                                  (print "answered late"))))
                       (t
                        (new-tone-sound 500 .5))))))
        ((equal *task* "Lexical")
         (cond ((equal *response* "z")
                (cond ((equal (cdr *current-lexical*) 'nonword)
                       (setf *time-elapsed* (- (mp-time) *start-time*))
                       (setf *nonword-rt* (+ *time-elapsed* *nonword-rt*))
                       (incf *number-nonword-trials*)
                       (incf *correct-nonword-trials*))
                      ((equal (cdr *current-lexical*) 'word)
                       (incf *number-word-trials*))
                      ((equal (cdr *current-lexical*) 'unsolved)
                       (incf *number-unsolved-trials*))
                      ((equal (cdr *current-lexical*) 'unseen)
                       (incf *number-unseen-trials*))
                      (t
                       (print "shouldn't be here")))
                (schedule-event-relative .25 'display-next-lexical))
               ((equal *response* "/")
                (cond ((equal (cdr *current-lexical*) 'nonword)
                       (incf *number-nonword-trials*))
                      ((equal (cdr *current-lexical*) 'word)
                       (setf *time-elapsed* (- (mp-time) *start-time*))
                       (setf *word-rt* (+ *time-elapsed* *word-rt*))
                       (incf *number-word-trials*)
                       (incf *correct-word-trials*))
                      ((equal (cdr *current-lexical*) 'unsolved)
                       (setf *time-elapsed* (- (mp-time) *start-time*))
                       (setf *unsolved-rt* (+ *time-elapsed* *unsolved-rt*))
                       (incf *number-unsolved-trials*)
                       (incf *correct-unsolved-trials*))
                      ((equal (cdr *current-lexical*) 'unseen)
                       (setf *time-elapsed* (- (mp-time) *start-time*))
                       (setf *unseen-rt* (+ *time-elapsed* *unseen-rt*))
                       (incf *number-unseen-trials*)
                       (incf *correct-unseen-trials*))
                      (t
                       (print "shouldn't be here")))
                (schedule-event-relative .25 'display-next-lexical))))
```

```
       (t
        (print "no matches"))))

(defun display-feedback (text)
  (clear-exp-window)
  (add-text-to-exp-window :x 150 :y 150 :text text)
  (proc-display)
  (schedule-event-relative 1 'display-next-problem))

(defun display-first-problem ()
  (set-hand-location left 4 4)
  (set-hand-location right 7 4)
  (reset-retrieval-record)
  (setf *current-problem* (pop *problems*))
  (print-chunk-activations)
  (setf *task* "RAT")
  (add-dm-fct (list (list 'RAT1Goal 'isa 'RATproblem 'status 'read-problem)))
  (goal-focus RAT1Goal)
  (build-initial-display (cadr *current-problem*) (caddr *current-problem*) (cadddr *current-problem*))
  (setf *end-event* (schedule-event-relative 30 'timeout))
  (setf *start-time* (mp-time))
  (run-full-time 2500))

(defun display-next-problem ()
  (cond ((> (length *problems*) 0)
        (setf *current-problem* (pop *problems*))
        (print-chunk-activations)
        (setf *end-event* (schedule-event-relative 30 'timeout))
        (setf *start-time* (mp-time))
        (build-display (cadr *current-problem*) (caddr *current-problem*) (cadddr *current-problem*)))
       ((equal *task* "RAT")
        (setf *task* "Lexical")
        (display-first-lexical))
       ((equal *task* "RAT2")
        (cond ((equal *current-block* 42)
)
          (t
             (setf *task* "Lexical")
             (display-first-lexical))))))

(defun safe-div (x y)
  (cond ((= y 0) nil)
       (t (/ x y))))

(defun print-results ()
  (push (list *unsolved*
          *correct*
          (safe-div *correct-sum* *correct*)
          (safe-div *already-solved* *correct*)
          (safe-div *already-solved-sum* *already-solved*)
          (safe-div *improvednh* (- *unsolved* *numberHints*))
          (safe-div *improvednh-sum* *improvednh*)
          (safe-div *improvedh* *numberHints*)
          (safe-div *improvedh-sum* *improvedh*)
```

```
                (/ *solvednh* 5)
                (safe-div *solvednh-sum* *solvednh*)
                (/ *solvedh* 5)
                (safe-div *solvedh-sum* *solvedh*)
                (safe-div *correct-unsolved-trials* *number-unsolved-trials*)
                (safe-div *unsolved-rt* *correct-unsolved-trials*)
                (/ *correct-unseen-trials* *number-unseen-trials*)
                (safe-div *unseen-rt* *correct-unseen-trials*)
                (/ *correct-word-trials* *number-word-trials*)
                (safe-div *word-rt* *correct-word-trials*)
                (/ *correct-nonword-trials* *number-nonword-trials*)
                (safe-div *nonword-rt* *correct-nonword-trials*))
            *results-list*)
  (print *unsolved*)
  (print *correct*)
  (print (safe-div *correct-sum* *correct*))
  (print (safe-div *already-solved* *correct*))
  (print (safe-div *already-solved-sum* *already-solved*))
  (print (safe-div *improvednh* (- *unsolved* *numberHints*)))
  (print (safe-div *improvednh-sum* *improvednh*))
  (print (safe-div *improvedh* *numberHints*))
  (print (safe-div *improvedh-sum* *improvedh*))
  (print (/ *solvednh* 5))
  (print (safe-div *solvednh-sum* *solvednh*))
  (print (/ *solvedh* 5))
  (print (safe-div *solvedh-sum* *solvedh*))
  (print (safe-div *correct-unsolved-trials* *number-unsolved-trials*))
  (print (safe-div *unsolved-rt* *correct-unsolved-trials*))
  (print (/ *correct-unseen-trials* *number-unseen-trials*))
  (print (safe-div *unseen-rt* *correct-unseen-trials*))
  (print (/ *correct-word-trials* *number-word-trials*))
  (print (safe-div *word-rt* *correct-word-trials*))
  (print (/ *correct-nonword-trials* *number-nonword-trials*))
  (print (safe-div *nonword-rt* *correct-nonword-trials*)))

(defun safe-average (results n)
  (let ((subjects n)
            (tempsum 0))
    (dolist (x results)
      (cond ((eql x nil)
               (decf subjects))
              (t
               (setf tempsum (+ x tempsum)))))
    (cond ((= subjects 0) nil)
             (t (/ tempsum subjects)))))

(defun average-results (n)
  (setf *average-results*
            (list
             (safe-average (mapcar #'(lambda (x) (nth 0 x)) *results-list*) n)
             (safe-average (mapcar #'(lambda (x) (nth 1 x)) *results-list*) n)
             (safe-average (mapcar #'(lambda (x) (nth 2 x)) *results-list*) n)
             (safe-average (mapcar #'(lambda (x) (nth 3 x)) *results-list*) n)
             (safe-average (mapcar #'(lambda (x) (nth 4 x)) *results-list*) n)
```

```
          (safe-average (mapcar #'(lambda (x) (nth 5 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 6 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 7 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 8 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 9 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 10 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 11 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 12 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 13 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 14 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 15 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 16 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 17 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 18 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 19 x)) *results-list*) n)
          (safe-average (mapcar #'(lambda (x) (nth 20 x)) *results-list*) n)))
(let ((initial nil)
          (solved nil)
          (unsolvednh nil)
          (unsolvedh nil)
          (unseennh nil)
          (unseenh nil))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 0 x)) *retrieval-results-list*)) n)
          initial))
  (setf initial (reverse initial))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 1 x)) *retrieval-results-list*)) n)
          solved))
  (setf solved (reverse solved))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 2 x)) *retrieval-results-list*)) n)
          unsolvednh))
  (setf unsolvednh (reverse unsolvednh))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 3 x)) *retrieval-results-list*)) n)
          unsolvedh))
  (setf unsolvedh (reverse unsolvedh))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 4 x)) *retrieval-results-list*)) n)
          unseennh))
  (setf unseennh (reverse unseennh))
  (dotimes (j 9)
    (push (safe-average (mapcar #'(lambda (y) (nth j y))
                                (mapcar #'(lambda (x) (nth 5 x)) *retrieval-results-list*)) n)
          unseenh))
  (setf unseenh (reverse unseenh))
  (setf *average-retrieval-results* (list initial solved unsolvednh unsolvedh unseennh unseenh))))
```

```
(defun print-retrieval-results ()
  (mapcar #'(lambda (x) (mapcar #'(lambda (y) (format nil "~4,2F" y)) x)) *average-retrieval-results*))

(defun print-average-results ()
  (print (mapcar #'(lambda (x) (format nil "~4,2F" x)) *average-results*)))

(defun display-first-lexical ()
  (set-hand-location left 1 5)
  (set-hand-location right 10 5)
  (clear-exp-window)
  (cond ((or (= *current-block* 12) (= *current-block* 22) (= *current-block* 32) (= *current-block* 42))
              (setf *lexical-items* nil)
              (setf *filler-words* (permute-list *filler-words*))
              (setf *nonwords* (permute-list *nonwords*))
              (dotimes (i 10)
                (push (cons (pop *filler-words*) 'word) *lexical-items*)
                (push (cons (pop *nonwords*) 'nonword) *lexical-items*))
              (setf *lexical-items* (permute-list *lexical-items*))
              (setf *current-lexical* (pop *lexical-items*))
              (add-text-to-exp-window :x 150 :y 150 :text (car *current-lexical*))
              (let ((new-goal (gentemp 'lexicalGoal)))
                (add-dm-fct (list (list new-goal 'isa 'lexicalDecision 'status 'read-string)))
                (goal-focus-fct new-goal))
              (setf *start-time* (mp-time))
              (proc-display))
             (t
              (cond (*answers* (setf *answers* (permute-list *answers*)))
                    (t nil))
              (setf *lexical-items* nil)
              (setf *not-presented* nil)
              (setf *presented* nil)
              (setf *unseen-presented* nil)
              (setf *unseen-not-presented* nil)
              (cond ((equal (length *answers*) 0))
                    ((equal (mod (length *answers*) 2) 1)
                          (cond ((equal *extra-hint* nil)
                                     (cond ((= 0 (act-r-random 2))
                                                 (setf *extra-hint* 1)
                                                 (push (car *answers*) *not-presented*)
                                                 (push (cons (pop *answers*) 0) *hintp-alist*)
                                                 (setf *numberHints* (+ *numberHints* (/ (length *answers*) 2))))
                                                (t
                                                 (setf *extra-hint* 0)
                                                 (push (car *answers*) *presented*)
                                                 (push (cons (pop *answers*) 1) *hintp-alist*)
                                                 (setf *numberHints* (+ *numberHints* (1+ (/ (length *answers*) 2)))))))
                                    ((= *extra-hint* 0)
                                     (setf *extra-hint* 1)
                                     (push (car *answers*) *not-presented*)
                                     (push (cons (pop *answers*) 0) *hintp-alist*)
                                     (setf *numberHints* (+ *numberHints* (/ (length *answers*) 2))))
                                    ((= *extra-hint* 1)
                                     (setf *extra-hint* 0)
                                     (push (car *answers*) *presented*)
```

```
                  (push (cons (pop *answers*) 1) *hintp-alist*)
                  (setf *numberHints* (+ *numberHints* (1+ (/ (length *answers*) 2)))))))
          (t
              (setf *numberHints* (+ *numberHints* (/ (length *answers*) 2)))))
        (dolist (x *hints*)
         (push x *unseen-presented*)
         (push (cons x -1) *hintp-alist*))
        (dolist (x *notHints*)
         (push x *unseen-not-presented*)
         (push (cons x -2) *hintp-alist*))
        (dotimes (i (/ (length *answers*) 2))
         (push (car *answers*) *presented*)
         (push (cons (pop *answers*) 1) *hintp-alist*)
         (push (car *answers*) *not-presented*)
         (push (cons (pop *answers*) 0) *hintp-alist*))
        (setf *filler-words* (permute-list *filler-words*))
        (setf *nonwords* (permute-list *nonwords*))
        (dotimes (i 10)
         (push (cons (pop *filler-words*) 'word) *lexical-items*)
         (push (cons (pop *nonwords*) 'nonword) *lexical-items*))
        (setf *lexical-items* (append (mapcar #'(lambda (x) (cons (string x) 'unsolved)) *presented*) *lexical-
items*))
        (setf *lexical-items* (append (mapcar #'(lambda (x) (cons (string x) 'unseen)) *unseen-presented*)
*lexical-items*))
        (dotimes (i (+ (length *presented*) (length *unseen-presented*)))
         (push (cons (pop *nonwords*) 'nonword) *lexical-items*))
        (setf *lexical-items* (permute-list *lexical-items*))
        (setf *current-lexical* (pop *lexical-items*))
        (add-text-to-exp-window :x 150 :y 150 :text (car *current-lexical*))
        (let ((new-goal (gentemp 'lexicalGoal)))
         (add-dm-fct (list (list new-goal 'isa 'lexicalDecision 'status 'read-string)))
         (setf *answers* nil)
         (goal-focus-fct new-goal))
        (setf *start-time* (mp-time))
        (proc-display))))

(defun display-next-lexical ()
 (clear-exp-window)
 (cond ((= (length *lexical-items*) 0)
        (reset-retrieval-record)
        (advance-block)
        (setf *current-problem* (pop *problems*))
        (set-hand-location left 4 4)
        (set-hand-location right 7 4)
        (let ((new-goal (gentemp 'RATGoal)))
         (add-dm-fct (list (list new-goal 'isa 'RATproblem 'status 'read-problem)))
         (goal-focus-fct new-goal))
        (setf *start-time* (mp-time))
        (setf *end-event* (schedule-event-relative 30 'timeout))
        (build-display (cadr *current-problem*) (caddr *current-problem*) (cadddr *current-problem*)))
       (t
        (setf *current-lexical* (pop *lexical-items*))
        (add-text-to-exp-window :x 150 :y 150 :text (car *current-lexical*))
        (setf *start-time* (mp-time))
```

```
        (proc-display))))

(defun timeout ()
 (cond ((equal *task* "RAT")
        (push (car *current-problem*) *answers*)
        (push *current-problem* *unsolved-problems*)
        (push (cons (car *current-problem*) 0) *solvedp-alist*)
        (setf *unsolved* (1+ *unsolved*))
        (print "timeout")
        (process-retrieval-record)
        (when *print-activations* (print "unsolved" *output*))
        (display-feedback "unsolved"))
       ((equal *task* "RAT2")
        (print "timeout")
        (process-retrieval-record)
        (display-feedback "unsolved"))))

(defun reset-retrieval-record ()
 (setf *retrieve-word-failure* 0)
 (setf *retrieve-solved-problem* 0)
 (setf *retrieve-compound-failure* 0)
 (setf *retrieve-new-related-word* 0)
 (setf *retrieve-old-related-word* 0)
 (setf *retrieve-unrelated-word* 0)
 (setf *retrieve-new-related-compound* 0)
 (setf *retrieve-old-related-compound* 0)
 (setf *retrieve-unrelated-compound* 0)
 (setf *retrieved-chunks* nil))

(defun record-retrieval (chunk)
 (cond ((equal chunk 'word-failure)
        (incf *retrieve-word-failure*)
        )
       ((equal chunk 'compound-failure)
        (incf *retrieve-compound-failure*)
        )
       ((equal chunk 'problem-recalled)
        (incf *retrieve-solved-problem*)
        )
       ((wordp chunk)
        (cond ((related-wordp chunk *current-problem*)
               (cond ((or (member chunk *retrieved-chunks* :test #'equal-chunks-fct)
                          (member chunk (cadr (assoc (nth 4 *current-problem*) *retrieved-chunks-list*)) :test
#'equal-chunks-fct))
                      (incf *retrieve-old-related-word*)
                      )
                     (t
                      (incf *retrieve-new-related-word*)
                      )))
           (t
               (incf *retrieve-unrelated-word*)
               (print "retrieved unrelated word")
               ))
        )
```

```
            ((compoundp chunk)
             (cond ((related-compoundp chunk *current-problem*)
                    (cond ((or (member chunk *retrieved-chunks* :test #'equal-chunks-fct)
                               (member chunk (cadr (assoc (nth 4 *current-problem*) *retrieved-chunks-list*)) :test
#'equal-chunks-fct))
                           (incf *retrieve-old-related-compound*)
                           )
                          (t
                           (incf *retrieve-new-related-compound*)
                           )))
                   (t
                    (incf *retrieve-unrelated-compound*)
                    (print "retrieved unrelated compound")
                    ))
             )
            (t
             (print "shouldn't be here in record-retrieval")
             (print chunk)
             (break)))
  (push chunk *retrieved-chunks*))

(defun process-retrieval-record ()
  (push (list (nth 4 *current-problem*) *retrieved-chunks*) *retrieved-chunks-list*)
  (let* ((denominator (+ *retrieve-word-failure* *retrieve-compound-failure*
                         *retrieve-new-related-word* *retrieve-old-related-word*
                         *retrieve-unrelated-word* *retrieve-new-related-compound*
                         *retrieve-old-related-compound* *retrieve-unrelated-compound*
                         *retrieve-solved-problem*))
             (recalled-problems (/ *retrieve-solved-problem* denominator))
             (word-failures (/ *retrieve-word-failure* denominator))
             (compound-failures (/ *retrieve-compound-failure* denominator))
             (new-related-words (/ *retrieve-new-related-word* denominator))
             (old-related-words (/ *retrieve-old-related-word* denominator))
             (unrelated-words (/ *retrieve-unrelated-word* denominator))
             (new-related-compounds (/ *retrieve-new-related-compound* denominator))
             (old-related-compounds (/ *retrieve-old-related-compound* denominator))
             (unrelated-compounds (/ *retrieve-unrelated-compound* denominator)))
    (cond ((equal *task* "RAT")
              (push (list recalled-problems word-failures compound-failures new-related-words old-related-words
                          unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                    *initial-retrievals*))
             ((equal *task* "RAT2")
              (cond ((equal (cdr (assoc (car *current-problem*) *solvedp-alist*)) 1)
                     (push (list recalled-problems word-failures compound-failures new-related-words old-related-
words
                          unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                             *solved-retrievals*))
                    ((equal (cdr (assoc (car *current-problem*) *solvedp-alist*)) 0)
                     (cond ((equal (cdr (assoc (car *current-problem*) *hintp-alist*)) 0)
                            (push (list recalled-problems word-failures compound-failures new-related-words old-
related-words
                          unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                                  *unsolvednh-retrievals*))
                           ((equal (cdr (assoc (car *current-problem*) *hintp-alist*)) 1)
```

```
                                  (push (list recalled-problems word-failures compound-failures new-related-words old-
related-words
                            unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                                   *unsolvedh-retrievals*))
                                 (t
                                  (print "shouldn't be here in process-retrieval-record 3"))))
                        ((equal (cdr (assoc (car *current-problem*) *hintp-alist*)) -1)
                         (push (list recalled-problems word-failures compound-failures new-related-words old-related-
words
                            unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                                 *unseenh-retrievals*))
                        ((equal (cdr (assoc (car *current-problem*) *hintp-alist*)) -2)
                         (push (list recalled-problems word-failures compound-failures new-related-words old-related-
words
                            unrelated-words new-related-compounds old-related-compounds unrelated-compounds)
                                 *unseennh-retrievals*))
                        (t
                         (print "shouldn't be here in process-retrieval-record 2")
                         (break))))
               (t
                (print "shouldn't be in another task in process-retrieval-record"))))
  (reset-retrieval-record))

(defun average-retrieval-records ()
  (let ((initial nil)
           (solved nil)
           (unsolvednh nil)
           (unsolvedh nil)
           (unseennh nil)
           (unseenh nil))
    (dotimes (i 9)
      (push (safe-average (mapcar #'(lambda (x) (nth i x)) *initial-retrievals*)
                                    (length *initial-retrievals*))
              initial))
    (setf initial (reverse initial))
    (dotimes (i 9)
      (push (safe-average (mapcar #'(lambda (x) (nth i x)) *solved-retrievals*)
                                    (length *solved-retrievals*))
              solved))
    (setf solved (reverse solved))
    (dotimes (i 9)
      (push (safe-average (mapcar #'(lambda (x) (nth i x)) *unsolvednh-retrievals*)
                                    (length *unsolvednh-retrievals*))
              unsolvednh))
    (setf unsolvednh (reverse unsolvednh))
    (dotimes (i 9)
      (push (safe-average (mapcar #'(lambda (x) (nth i x)) *unsolvedh-retrievals*)
                                    (length *unsolvedh-retrievals*))
              unsolvedh))
    (setf unsolvedh (reverse unsolvedh))
    (dotimes (i 9)
      (push (safe-average (mapcar #'(lambda (x) (nth i x)) *unseennh-retrievals*)
                                    (length *unseennh-retrievals*))
              unseennh))
```

```
   (setf unseennh (reverse unseennh))
   (dotimes (i 9)
     (push (safe-average (mapcar #'(lambda (x) (nth i x)) *unseenh-retrievals*)
                                     (length *unseenh-retrievals*))
               unseenh))
   (setf unseenh (reverse unseenh))
   (push (list initial solved unsolvednh unsolvedh unseennh unseenh) *retrieval-results-list*)))


(defun wordp (chunk)
  (cond ((equal (chunk-chunk-type-fct chunk) 'WORD) t)
           (t nil)))

(defun compoundp (chunk)
  (cond ((equal (chunk-chunk-type-fct chunk) 'COMPOUND-WORD) t)
           (t nil)))

(defun related-wordp (word problem)
  (let ((found nil)
           (answer (car problem))
           (meaning1 (chunk-slot-value-fct word 'meaning1))
           (meaning2 (chunk-slot-value-fct word 'meaning2))
           (meaning3 (chunk-slot-value-fct word 'meaning3)))
   (cond ((equal (chunk-slot-value-fct answer 'meaning1) meaning1)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning2) meaning1)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning3) meaning1)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning1) meaning2)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning2) meaning2)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning3) meaning2)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning1) meaning3)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning2) meaning3)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning3) meaning3)
             (setf found t)))
   (cond (found t)
            (t nil))))

(defun related-compoundp (compound problem)
  (let ((found nil)
           (answer (car problem))
           (meaning (chunk-slot-value-fct compound 'meaning)))
   (cond ((equal (chunk-slot-value-fct answer 'meaning1) meaning)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning2) meaning)
             (setf found t))
            ((equal (chunk-slot-value-fct answer 'meaning3) meaning)
             (setf found t)))
```

```
    (cond (found t)
              (t nil))))

(defun related-wordp2 (meaning problem)
  (let ((found nil)
            (answer (car problem)))
    (cond ((equal (chunk-slot-value-fct answer 'meaning1) meaning)
               (print (chunk-slot-value-fct answer 'meaning1))
               (print "meaning1")
               (setf found t))
              ((equal (chunk-slot-value-fct answer 'meaning2) meaning)
               (print (chunk-slot-value-fct answer 'meaning2))
               (print "meaning2")
               (setf found t))
              ((equal (chunk-slot-value-fct answer 'meaning3) meaning)
               (print (chunk-slot-value-fct answer 'meaning3))
               (print "meaning3")
               (setf found t)))
    (cond (found t)
              (t nil))))

(clear-all)

(define-model RAT

(sgp :v nil :esc t :show-focus t :trace-detail low :rt .4 :mas 9 :act nil
     :er t :bll .5 :ans .26 :ol 20 :lf .35 :declarative-num-finsts 3
     :declarative-finst-span 2.5 :egs 3 :ut -100)

(chunk-type word-mapping word text)
(chunk-type word meaning1 meaning2 meaning3 self-ref problem-ref)
(chunk-type compound-word comp1 comp2 value meaning)
(chunk-type RATproblem word1 word2 word3 answer status test1 test2 test3 temp current-concept compound1-
meaning compound2-meaning compound3-meaning done-before current-start)
(chunk-type lexicalDecision word text status)
(chunk-type meaning)

(p find-first-word
   =goal>
    isa RATproblem
    word1 nil
    status read-problem
   =visual-location>
    isa visual-location
==>
   +visual-location>
    isa visual-location
    :attended nil
    screen-y lowest
   =goal>
    status attend-word
)

(p attend-word
```

```
  =goal>
   isa RATproblem
   status attend-word
  =visual-location>
   isa visual-location
  ?visual>
   state free
==>
  +visual>
   isa move-attention
   screen-pos =visual-location
  =goal>
   status read-word
)

(p read-word
  =goal>
   isa RATproblem
   temp nil
   status read-word
  =visual>
   isa text
   value =word
  ?retrieval>
   state free
==>
  =goal>
   temp =word
   status read-word
  +retrieval>
   isa word-mapping
   text =word
)

(p access-meaning1
  =goal>
   isa RATproblem
   word1 nil
   temp =word
   status read-word
  =retrieval>
   isa word-mapping
   text =word
   word =meaning
==>
  =goal>
   temp nil
   word1 =meaning
   status read-problem
)

(p access-meaning2
  =goal>
   isa RATproblem
```

```
    word1 =word1
    word2 nil
    temp =word
    status read-word
   =retrieval>
    isa word-mapping
    text =word
    word =meaning
==>
   =goal>
    temp nil
    word2 =meaning
    status read-problem
)

(p access-meaning3
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 nil
    temp =word
    status read-word
   =retrieval>
    isa word-mapping
    text =word
    word =meaning
==>
   =goal>
    temp nil
    word3 =meaning
    status read-problem
)

(p find-second-word
   =goal>
    isa RATproblem
    word1 =word1
    word2 nil
    status read-problem
==>
   +visual-location>
    isa visual-location
    :attended nil
    screen-y lowest
   =goal>
    status attend-word
)

(p find-third-word
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
```

```
   word3 nil
   status read-problem
==>
  +visual-location>
   isa visual-location
   :attended nil
   screen-y lowest
  =goal>
   status attend-word
)

(p problem-encoded1
  =goal>
   isa RATproblem
   word1 =word1
   word2 =word2
   word3 =word3
   status read-problem
  !eval! (equal *task* "RAT")
==>
  =goal>
   temp nil
   status unsolved
)

(p problem-encoded2
  =goal>
   isa RATproblem
   word1 =word1
   word2 =word2
   word3 =word3
   status read-problem
  !eval! (equal *task* "RAT2")
==>
  =goal>
   temp nil
   status try-recall
)

(p recall-rat-initial
  =goal>
   isa RATproblem
   word1 =word1
   word2 =word2
   word3 =word3
   status try-recall
  ?retrieval>
   state free
==>
  +retrieval>
   isa RATproblem
   word1 =word1
   word2 =word2
   word3 =word3
```

```
  - answer nil
   =goal>
    status trying-recall
)

(p recall-rat
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    status search-concept
   ?retrieval>
    state free
   !eval! (equal *task* "RAT2")
==>
   +retrieval>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
   =goal>
    status trying-recall
)

(p recall-rat-success
   =goal>
    isa RATproblem
    answer nil
    status trying-recall
   =retrieval>
    isa RATproblem
    answer =answer
    status solved
==>
   =goal>
    answer =answer
    temp 5
    status type-answer
   !eval! (record-retrieval 'problem-recalled)
)

(p recall-rat-unsolved
   =goal>
    isa RATproblem
    answer nil
    status trying-recall
   =retrieval>
    isa RATproblem
    status unsolved
==>
   =goal>
    done-before no
    status unsolved
```

```
)

(p recall-rat-failure
   =goal>
    isa RATproblem
    status trying-recall
   ?retrieval>
    state error
==>
   =goal>
    status unsolved
)

(p start-with-first
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    temp nil
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (act-r-random 3)
   +retrieval> =word1
   =goal>
    temp =value
    current-start first
    status retrieving-word
)

(p start-with-second
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    temp nil
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (act-r-random 3)
   +retrieval> =word2
   =goal>
    temp =value
    current-start second
    status retrieving-word
)

(p start-with-third
   =goal>
    isa RATproblem
```

```
    word1 =word1
    word2 =word2
    word3 =word3
    temp nil
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (act-r-random 3)
   +retrieval> =word3
   =goal>
    temp =value
    current-start third
    status retrieving-word
)

;; make first slightly preferred
(spp start-with-second :p .8)
(spp start-with-third :p .8)

(p continue-with-first
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    temp =temp
    current-start first
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (cond ((= 0 =temp) nil) (t (1- =temp)))
   +retrieval> =word1
   =goal>
    temp =value
    status retrieving-word
)

(p continue-with-second
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    temp =temp
    current-start second
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (cond ((= 0 =temp) nil) (t (1- =temp)))
   +retrieval> =word2
   =goal>
```

```
    temp =value
    status retrieving-word
)

(p continue-with-third
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    temp =temp
    current-start third
    status unsolved
   ?retrieval>
    state free
==>
   !bind! =value (cond ((= 0 =temp) nil) (t (1- =temp)))
   +retrieval> =word3
   =goal>
    temp =value
    status retrieving-word
)

(p encode-search-concept
   =goal>
    isa RATproblem
    status retrieving-word
   =retrieval>
    isa word
    meaning1 =meaning
==>
   =goal>
    current-concept =meaning
    status search-concept
)

(p failed-find-search-concept
   =goal>
    isa RATproblem
    status retrieving-word
   ?retrieval>
    state error
==>
   =goal>
    temp nil
    current-start nil
    status unsolved
   !eval! (print "failed to retrieve concept")
)

(p retrieve-related-word
   =goal>
    isa RATproblem
    word1 =word1
```

```
    word2 =word2
    word3 =word3
    current-concept =meaning
    status search-concept
   ?retrieval>
    state free
==>
   !bind! =problem-ref (nth 4 *current-problem*)
   +retrieval>
    isa word
  - self-ref =word1
  - self-ref =word2
  - self-ref =word3
    problem-ref =problem-ref
    :recently-retrieved nil
   =goal>
    status retrieving-related-word
)

(p initial-failure-word
   =goal>
    isa RATproblem
    status retrieving-related-word
   ?retrieval>
    state error
==>
   =goal>
    current-concept nil
    status unsolved
   !eval! (record-retrieval 'word-failure)
)

(p initial-failure-compound
   =goal>
    isa RATproblem
    status retrieving-related-compound
   ?retrieval>
    state error
==>
   =goal>
    current-concept nil
    status unsolved
   !eval! (record-retrieval 'compound-failure)
)

(p found-related-word
   =goal>
    isa RATproblem
    current-concept =meaning
    status retrieving-related-word
   =retrieval>
    isa word
    self-ref =word
==>
```

```
  =goal>
   answer =word
   status test-found-word
  !eval! (record-retrieval =word)
)

(p word-test1-forward
  =goal>
   isa RATproblem
   word1 =word1
   answer =word
   test1 nil
   status test-found-word
  ?retrieval>
   state free
==>
  +retrieval>
   isa compound-word
   comp1 =word
   comp2 =word1
  =goal>
   status test-word1-forward
)

(p word-test1-backward
  =goal>
   isa RATproblem
   word1 =word1
   answer =word
   test1 nil
   status test-word1-forward
  ?retrieval>
   state error
==>
  +retrieval>
   isa compound-word
   comp1 =word1
   comp2 =word
  =goal>
   status test-word1-backward
)

(p failed-word-test1
  =goal>
   isa RATproblem
   answer =word
   status test-word1-backward
  ?retrieval>
   state error
==>
  +retrieval> =word
  =goal>
   test1 nil
   test2 nil
```

```
    test3 nil
    answer nil
    compound1-meaning nil
    compound2-meaning nil
    compound3-meaning nil
    status unsolved
)

(p test1-forward-success
   =goal>
    isa RATproblem
    word1 =word1
    answer =word
    status test-word1-forward
   =retrieval>
    isa compound-word
    comp1 =word
    comp2 =word1
    meaning =meaning
==>
   +retrieval> =word
   =goal>
    test1 t
    compound1-meaning =meaning
    status test-found-word
)

(p test1-backward-success
   =goal>
    isa RATproblem
    word1 =word1
    answer =word
    status test-word1-backward
   =retrieval>
    isa compound-word
    comp1 =word1
    comp2 =word
    meaning =meaning
==>
   +retrieval> =word
   =goal>
    test1 t
    compound1-meaning =meaning
    status test-found-word
)

(p word-test2-forward
   =goal>
    isa RATproblem
    word2 =word2
    answer =word
    test2 nil
    status test-found-word
   ?retrieval>
```

```
    state free
==>
   +retrieval>
    isa compound-word
    comp1 =word
    comp2 =word2
   =goal>
    status test-word2-forward
)

(p word-test2-backward
   =goal>
    isa RATproblem
    word2 =word2
    answer =word
    test2 nil
    status test-word2-forward
   ?retrieval>
    state error
==>
   +retrieval>
    isa compound-word
    comp1 =word2
    comp2 =word
   =goal>
    status test-word2-backward
)

(p failed-word-test2
   =goal>
    isa RATproblem
    answer =word
    status test-word2-backward
   ?retrieval>
    state error
==>
   +retrieval> =word
   =goal>
    test1 nil
    test2 nil
    test3 nil
    answer nil
    compound1-meaning nil
    compound2-meaning nil
    compound3-meaning nil
    status unsolved
)

(p test2-forward-success
  =goal>
    isa RATproblem
    word2 =word2
    answer =word
    status test-word2-forward
```

```
  =retrieval>
   isa compound-word
   comp1 =word
   comp2 =word2
   meaning =meaning
==>
  +retrieval> =word
  =goal>
   test2 t
   compound2-meaning =meaning
   status test-found-word
)

(p test2-backward-success
  =goal>
   isa RATproblem
   word2 =word2
   answer =word
   status test-word2-backward
  =retrieval>
   isa compound-word
   comp1 =word2
   comp2 =word
   meaning =meaning
==>
  +retrieval> =word
  =goal>
   test2 t
   compound2-meaning =meaning
   status test-found-word
)

(p word-test3-forward
  =goal>
   isa RATproblem
   word3 =word3
   answer =word
   test3 nil
   status test-found-word
  ?retrieval>
   state free
==>
  +retrieval>
   isa compound-word
   comp1 =word
   comp2 =word3
  =goal>
   status test-word3-forward
)

(p word-test3-backward
  =goal>
   isa RATproblem
   word3 =word3
```

```
    answer =word
    test3 nil
    status test-word3-forward
   ?retrieval>
    state error
==>
   +retrieval>
    isa compound-word
    comp1 =word3
    comp2 =word
   =goal>
    status test-word3-backward
)

(p failed-word-test3
   =goal>
    isa RATproblem
    answer =word
    status test-word3-backward
   ?retrieval>
    state error
==>
   +retrieval> =word
   =goal>
    test1 nil
    test2 nil
    test3 nil
    answer nil
    compound1-meaning nil
    compound2-meaning nil
    compound3-meaning nil
    status unsolved
)

(p test3-forward-success
   =goal>
    isa RATproblem
    word3 =word3
    answer =word
    status test-word3-forward
   =retrieval>
    isa compound-word
    comp1 =word
    comp2 =word3
    meaning =meaning
==>
   +retrieval> =word
   =goal>
    test3 t
    compound3-meaning =meaning
    status test-found-word
)

(p test3-backward-success
```

```
  =goal>
   isa RATproblem
   word3 =word3
   answer =word
   status test-word3-backward
  =retrieval>
   isa compound-word
   comp1 =word3
   comp2 =word
   meaning =meaning
==>
  +retrieval> =word
  =goal>
   test3 t
   compound3-meaning =meaning
   status test-found-word
)

(p found-answer-word
  =goal>
   isa RATproblem
   test1 t
   test2 t
   test3 t
   status test-found-word
   answer =answer
==>
  =goal>
   status type-answer
   temp 5
  !eval! (print "solved using word")
)

(p found-compound-word1
  =goal>
   isa RATproblem
   word1 =comp1
   current-concept =meaning
   status retrieving-related-compound
  =retrieval>
   isa compound-word
   comp1 =comp1
   comp2 =comp2
   meaning =meaning
==>
  =goal>
   answer =comp2
   test1 t
   compound1-meaning =meaning
   status testing
  !eval! (record-retrieval =retrieval)
)

(p found-compound-word2
```

```
  =goal>
   isa RATproblem
   word2 =comp1
   current-concept =meaning
   status retrieving-related-compound
  =retrieval>
   isa compound-word
   comp1 =comp1
   comp2 =comp2
   meaning =meaning
==>
  =goal>
   answer =comp2
   test2 t
   compound2-meaning =meaning
   status testing
  !eval! (record-retrieval =retrieval)
)

(p found-compound-word3
  =goal>
   isa RATproblem
   word3 =comp1
   current-concept =meaning
   status retrieving-related-compound
  =retrieval>
   isa compound-word
   comp1 =comp1
   comp2 =comp2
   meaning =meaning
==>
  =goal>
   answer =comp2
   test3 t
   compound3-meaning =meaning
   status testing
  !eval! (record-retrieval =retrieval)
)

(p found-compound-word1-2
  =goal>
   isa RATproblem
   word1 =comp1
   current-concept =meaning
   status retrieving-related-compound
  =retrieval>
   isa compound-word
   comp1 =comp2
   comp2 =comp1
   meaning =meaning
==>
  =goal>
   answer =comp2
   test1 t
```

```
    compound1-meaning =meaning
    status testing
   !eval! (record-retrieval =retrieval)
)

(p found-compound-word2-2
   =goal>
    isa RATproblem
    word2 =comp1
    current-concept =meaning
    status retrieving-related-compound
   =retrieval>
    isa compound-word
    comp1 =comp2
    comp2 =comp1
    meaning =meaning
==>
   =goal>
    answer =comp2
    test2 t
    compound2-meaning =meaning
    status testing
   !eval! (record-retrieval =retrieval)
)

(p found-compound-word3-2
   =goal>
    isa RATproblem
    word3 =comp1
    current-concept =meaning
    status retrieving-related-compound
   =retrieval>
    isa compound-word
    comp1 =comp2
    comp2 =comp1
    meaning =meaning
==>
   =goal>
    answer =comp2
    test3 t
    compound3-meaning =meaning
    status testing
   !eval! (record-retrieval =retrieval)
)

(p found-compound-word-none
   =goal>
    isa RATproblem
    word1 =word1
    word2 =word2
    word3 =word3
    current-concept =meaning
    status retrieving-related-compound
   =retrieval>
```

```
    isa compound-word
  - comp1 =word1
  - comp1 =word2
  - comp1 =word3
  - comp2 =word1
  - comp2 =word2
  - comp2 =word3
==>
   =goal>
    answer nil
    status unsolved
   !eval! (record-retrieval =retrieval)
)

(p test-word1
   =goal>
    isa RATproblem
    status testing
    word1 =word1
    test1 nil
    answer =answer
   ?retrieval>
    state free
==>
   =goal>
    status testing1
   +retrieval>
    isa compound-word
    comp1 =word1
    comp2 =answer
)

(p test-word2
   =goal>
    isa RATproblem
    status testing
    word2 =word2
    test2 nil
    answer =answer
   ?retrieval>
    state free
==>
   =goal>
    status testing2
   +retrieval>
    isa compound-word
    comp1 =word2
    comp2 =answer
)

(p test-word3
   =goal>
    isa RATproblem
    status testing
```

```
    word3 =word3
    test3 nil
    answer =answer
   ?retrieval>
    state free
==>
   =goal>
    status testing3
   +retrieval>
    isa compound-word
    comp1 =word3
    comp2 =answer
)

(p found-test-word1
   =goal>
    isa RATproblem
    word1 =word1
    answer =answer
    test1 nil
    status testing1
   =retrieval>
    isa compound-word
    comp1 =word1
    comp2 =answer
    meaning =meaning
==>
   =goal>
    test1 t
    compound1-meaning =meaning
    status testing
)

(p found-test-word2
   =goal>
    isa RATproblem
    word2 =word2
    answer =answer
    test2 nil
    status testing2
   =retrieval>
    isa compound-word
    comp1 =word2
    comp2 =answer
    meaning =meaning
==>
   =goal>
    test2 t
    compound2-meaning =meaning
    status testing
)

(p found-test-word3
   =goal>
```

```
    isa RATproblem
    word3 =word3
    answer =answer
    test3 nil
    status testing3
  =retrieval>
    isa compound-word
    comp1 =word3
    comp2 =answer
    meaning =meaning
==>
  =goal>
    test3 t
    compound3-meaning =meaning
    status testing
)

(p failed-recall-test-word1
  =goal>
    isa RATproblem
    answer =word
    status testing1
  ?retrieval>
    state error
==>
  +retrieval> =word
  =goal>
    status unsolved
    answer nil
    test1 nil
    test2 nil
    test3 nil
    compound1-meaning nil
    compound2-meaning nil
    compound3-meaning nil
)

(p failed-recall-test-word2
  =goal>
    isa RATproblem
    answer =word
    status testing2
  ?retrieval>
    state error
==>
  +retrieval> =word
  =goal>
    status unsolved
    answer nil
    test1 nil
    test2 nil
    test3 nil
    compound1-meaning nil
    compound2-meaning nil
```

```
   compound3-meaning nil
)

(p failed-recall-test-word3
  =goal>
   isa RATproblem
   answer =word
   status testing3
  ?retrieval>
   state error
==>
  +retrieval> =word
  =goal>
   status unsolved
   answer nil
   test1 nil
   test2 nil
   test3 nil
   compound1-meaning nil
   compound2-meaning nil
   compound3-meaning nil
)

(p answer-found
  =goal>
   isa RATproblem
   status testing
   test1 t
   test2 t
   test3 t
   answer =answer
==>
  =goal>
   status type-answer
   temp 5
  !eval! (print "solved using compound")
)

(p type-answer
  =goal>
   isa RATproblem
   status type-answer
   temp =num
 - temp 0
  ?manual>
   state free
==>
  !bind! =newnum (- =num 1)
  +manual>
   isa press-key
   key "a"
  =goal>
   temp =newnum
)
```

```
(p type-enter
   =goal>
    isa RATproblem
    status type-answer
    answer =answer
    temp 0
   ?manual>
    state free
==>
   +manual>
    isa press-key
    key return
   =goal>
    status done
   !output! =answer
   !eval! (setf *answer* =answer)
)

(p display-cleared
   =goal>
    isa RATproblem
  - status attend-word
  - status read-problem
  - status encode-feedback
   =visual-location>
    isa visual-location
   ?visual>
    state free
==>
   =goal>
    status encode-feedback
   +visual>
    isa move-attention
    screen-pos =visual-location
)

(p encode-solved-feedback
   =goal>
    isa RATproblem
    status encode-feedback
   =visual>
    isa text
    value "solved"
==>
   =goal>
    status solved
   +goal>
    isa RATproblem
    status read-problem
)

(p encode-unsolved-feedback
   =goal>
```

```
    isa RATproblem
    status encode-feedback
   =visual>
    isa text
    value "unsolved"
==>
   =goal>
    answer nil
    status unsolved
   +goal>
    isa RATproblem
    status read-problem
)

(p attend-string
   =goal>
    isa lexicalDecision
    status read-string
   =visual-location>
    isa visual-location
   ?visual>
    state free
==>
   +visual>
    isa move-attention
    screen-pos =visual-location
   =goal>
    status read-word
)

(p read-string
   =goal>
    isa lexicalDecision
    text nil
    status read-word
   =visual>
    isa text
    value =word
   ?retrieval>
    state free
==>
   +retrieval>
    isa word-mapping
    text =word
   =goal>
    text =word
    status access-meaning
)

(p access-string-meaning
   =goal>
    isa lexicalDecision
    word nil
    text =text
```

```
    status access-meaning
   =retrieval>
    isa word-mapping
    text =text
    word =meaning
==>
   =goal>
    word =meaning
    status retrieve-word
)

(p access-string-word
   =goal>
    isa lexicalDecision
    word =word
    status retrieve-word
   ?retrieval>
    state free
==>
   +retrieval> =word
   =goal>
    status press-key
)

(p press-word
   =goal>
    isa lexicalDecision
    status press-key
   =retrieval>
    isa word
   ?manual>
    state free
==>
   =goal>
    status done
   +manual>
    isa punch
    hand right
    finger index
   +goal>
    isa lexicalDecision
    status read-string
)

(p press-nonword-no-meaning
   =goal>
    isa lexicalDecision
    status access-meaning
   ?retrieval>
    state error
   ?manual>
    state free
==>
   =goal>
```

```
    status done
   +manual>
    isa punch
    hand left
    finger index
   +goal>
    isa lexicalDecision
    status read-string
)

(p press-nonword
   =goal>
    isa lexicalDecision
    status press-key
   ?retrieval>
    state error
   ?manual>
    state free
==>
   =goal>
    status done
   +manual>
    isa punch
    hand left
    finger index
   +goal>
    isa lexicalDecision
    status read-string
)

(p detected-sound
   =aural-location>
    isa audio-event
    attended nil
   ?aural>
    state free
==>
   +aural>
    isa sound
    event =aural-location
)

(p sound-unsolved
   =goal>
    isa RATproblem
   =aural>
    isa sound
==>
   =goal>
    answer nil
    status unsolved
)

)
```