

## Representation of memory for order of mental operations in cognitive tasks

---

MICHAEL K. GARDNER, DAN J. WOLTZ, AND  
BRIAN G. BELL  
University of Utah



*The American Journal of Psychology*  
Summer 2002, Vol. 115, No. 2, pp. 251–274

---

Content in the *AJP* database is intended for personal, noncommercial use only. You may not reproduce, publish, distribute, transmit, participate in the transfer or sale of, modify, create derivative works from, display, or in any way exploit the *AJP* content in whole or in part without the written permission of the copyright holder.

---

To request permission to reprint material from *The American Journal of Psychology*, please find us online at:

<http://www.press.uillinois.edu/about/permission.html>

or email us at:

UIP-RIGHTS@uillinois.edu

© 2002 by the Board of Trustees of the University of Illinois

# Representation of memory for order of mental operations in cognitive tasks

---

MICHAEL K. GARDNER, DAN J. WOLTZ, AND BRIAN G. BELL  
University of Utah

Recent research shows that people learning a cognitive task acquire a memory for the order of operations applied, independent of the data to which those operations were applied. We designed two experiments to show how this sequence memory is represented. Experiment 1 compared predictions based on 3 possible sequence representation methods: composition, dyad transition, and associative chain. Latency and error results from a simple sequential task supported the associative chain representation. The associative links between operations presumably enhance performance by priming subsequent operations but do not operate in an all-or-none fashion. Experiment 2 explored whether transfer items that matched the first 2 rules and first 3 elements of a training item could bias participants toward executing a composed production learned during training. Latency and undetected error results were consistent with an associative chain representation but not with additional predictions made by the composition representation. These two experiments support the representation of operation sequences in memory as an associative chain.

An important class of cognitive skills involves the sequential application of a set of rules or elementary procedures. For example, when we solve an algebra equation, we apply the operations of addition, subtraction, multiplication, and division in a particular order. Different equations entail different orderings of the basic operations, but the operations themselves remain the same. Similarly, speech articulation involves using a set of basic grammatical rules to produce a surface structure consistent with the intended deep structure or meaning. These grammatical rules are applied in different orders to produce different sentences, but the basic rules remain the same over all utterances in a given language. Other examples abound. Skilled cooks use a set of cooking rules in preparing dishes; the rules involve things such as how to sweeten a mixture or how to combine a powder ingredient with a liquid. The rules are used in different orders when preparing different meals.

In the tasks just described, the sequences of rules can operate on different data across occasions that result in different outcomes or responses. For example, the equations  $3 + (4 \times 2)$  and  $5 + (7 \times 4)$  both involve multiplication followed by addition. The actual numbers mul-

tiplied and added differ in the two equations, which results in different outcomes or responses. Not all sequential rule-based tasks have this character. A standard checklist reviewed by a flight engineer or pilot before takeoff contains consistent data (i.e., the same items to be checked in the same order), and the rules or operations result in the same responses or outcomes unless a problem is encountered. Similarly, in tasks such as learning a word processor, one learns keystrokes to accomplish certain goals such as indenting or deleting. In this case the rules and the actions are inseparable. To delete a word you must not only depress keys in a certain order but must also depress particular keys. In experimental research on skill acquisition, the serial reaction time task developed by Nissen and Bullemer (1987), which has been used extensively to research sequence learning, represents an example of consistency in the mapping of rules to data and responses.

In sequential skills, learning is facilitated when there is a consistent mapping of rules to data and responses, in much the same way that consistency facilitates movement toward automaticity in any skill (e.g., Ackerman, 1988; Logan, 1988; Schneider & Shiffrin, 1977; Shiffrin & Schneider, 1977). However, our interest is in tasks in which the rules or operations can operate on different data on different occasions. These tasks usually involve more complex information processing, are more typical of educational or training situations, and are less well understood in terms of the theoretical mechanisms involved in skill acquisition.

Lundy, Wegner, Schmidt, and Carlson (1994) made an important distinction between multistep sequential cognitive tasks that operate on variable data. They noted that in some multistep tasks, the output of one step becomes the input for a later step. Lundy et al. called these steps cascaded component steps. In other multistep tasks, the output of a given step is not needed in future steps. They called these steps encapsulated component steps. The distinction is important because Lundy et al. found that the benefits of practice with a consistent sequence held only for tasks with cascaded component steps. In the research we present here, we focused on a task with cascaded component steps. Although this may be seen as a limitation on the generality of findings on sequence memory, one should keep in mind that many complex mental skills are of the cascaded variety (e.g., diagnosis, calculation, and programming).

### **What accounts for learning in sequential processing tasks?**

Simple sequential learning, such as that demonstrated in the serial reaction time task (Nissen & Bullemer, 1987), has been studied exten-

sively. In this task, several findings seem apparent. First, learning ambiguous sequences (in which a given item is not uniquely followed by another item) seems to entail attention (Cohen, Ivry, & Keele, 1990). Second, people appear to be able to learn sequences of great length if the sequences do not contain a high degree of ambiguity (Stadler & Neely, 1997). Third, performance in this task may be particularly related to spatial learning rather than symbolic or other types of learning (Koch & Hoffmann, 2000).

In contrast, learning has also been studied in more complex sequential tasks, including those in which processing operations and data are independent. Far less is known about the nature and representation of sequential learning in these tasks, and such tasks seem to be fundamentally different from the serial reaction time task (e.g., they involve general transformation rules, cascaded operations, and multiple sequences of operations within the same task). We now outline some of the mechanisms that may be responsible for sequential learning in more complex task environments.

Perhaps the simplest explanation for learning in complex sequential tasks is that participants represent only the rules or operations in memory. The basic rules can then be ordered in different ways depending on the problems encountered. According to this explanation, performance improvements with practice are the result of greater processing efficiency in applying or retrieving the rules or operations (i.e., they take less attention to execute or have greater memory strength). Schneider (1985) proposed a theory of skill acquisition consistent with this explanation, and Carlson, Sullivan, and Schneider (1989) found empirical support for the component-strengthening hypothesis in a study involving the learning of logic gates.

Other theories are also consistent with the component rule or operation-based explanation of skill acquisition. Logan (1988) described a class of theories that he labeled as process based (e.g., LaBerge & Samuels, 1974; Logan, 1978). Like Schneider (1985), these theories emphasize changes in attention mechanisms more than a reorganization of the underlying memory representation. Also, some theories have proposed strengthening of the underlying rules or operations as one of several memory mechanisms underlying skill acquisition in a task (e.g., Anderson, 1983, 1987, 1993; MacKay, 1982, 1987).

A second explanation of skill acquisition in sequential processing tasks posits that people rely on memory for individual instances they have encountered. The notion here is that initial, effortful algorithmic approaches to task solution are supplanted by effortless memory retrieval of individual instances and their solutions. Whereas some theories (e.g.,

Anderson, 1983, 1987, 1993) have incorporated instance memory as one of a number of factors involved in skill acquisition, Logan (1988) proposed that instance memory alone was sufficient to account for advanced performance in many tasks. Carlson and Lundy (1992) also found evidence of data-specific representation underlying a more complex mental computation task. Although most researchers would agree that individual instances play some role in performance, the strong position that memory for individual instances can account for all of skill acquisition seems somewhat less likely.

A third explanation of skill acquisition involves a memory for the order of operations performed during the acquisition of a skill. This explanation posits that people store not only the component operations or rules and the surface structure of individual instances but also a more abstract representation that contains the order in which individual operations were performed. This memory representation can be data general; that is, it can be independent of the actual data that the operations in question are transforming (depending on the task demands imposed during training). Whereas empirical support for the first two explanations of skill acquisition has existed for some time, support for the third explanation has emerged recently, except for Luchins's (1942) early demonstration of the *Einstellung* effect in problem solving.

McKendree and Anderson (1987) found that people evaluating LISP expressions (function combinations) were faster and more accurate when evaluating expressions they had seen more frequently in the past, despite the fact that their previous encounters with these expressions involved different data. Frensch (1991) found that when participants were given equal amounts of practice on individual steps in a multistep computation problem, those who had practiced the steps in random order during training were slower during transfer than those who had practiced the steps in a canonical order consistent with transfer. Carlson and Lundy (1992), in a mental arithmetic task, found that sequence and data consistency had separable effects. Their participants benefited from consistent sequences of operations even when data values varied. In another calculation task, Lundy et al. (1994) found benefit for consistent sequences regardless of whether the task had a hierarchical or flat goal structure. Wegner and Carlson (1996) also found some support for a benefit from consistent transitions between mental operations in a lengthy (12-step) arithmetic calculation task. The benefit in this study was limited to conditions with simple subgoal structure. Finally, Woltz, Bell, Kyllonen, and Gardner (1996; also see Woltz, Gardner, & Bell, 2000) reported that participants applying simple rules to reduce strings of digits to single-digit responses were faster and less error prone for sequences of operations they had previously encountered.

### How is sequence information represented in memory?

Given that people seem to benefit from a sequence of operations previously encountered, how is this information represented in memory?

**Composition or chunking.** One possibility is that during learning several operations become fused together into a single multioperation chain, which acts in an all-or-none fashion. Such a representation was posited by Anderson (1983), and he called it composition. Some positive evidence of composition has been reported (Frensch, 1991), but there also has been criticism of this mechanism (Carlson & Schneider, 1989; Carlson et al., 1989). In addition, Anderson dropped the composition mechanism from his later version of the adaptive control of thought (ACT) theory (Anderson, 1993).

**Rule transitions.** Another possibility is that the implicit learning of sequences of operations is based in the learning of operation pairs or dyads. That is, people learn a transition from one operation to another. Thus, in learning the sequence of operations A–B–C, people learn the individual pairings or transitions A–B and B–C, presumably through a mechanism such as temporal contiguity. However, they do not learn the sequence A–B–C as a whole. Related to this, Wegner and Carlson (1996) noted that ability to benefit from consistent transitions probably depends on the ability to build unique associations between pairs of operators.

Depending on the level of abstraction at which this information is stored, the learning of transition dyads (A–B and B–C) may or may not be sensitive to position of occurrence within the sequence. In the first case, the operation dyad A–B would facilitate performance only on sequences containing the same dyad in the first position (e.g., facilitation for the sequence A–B–D but not D–A–B). In the second case, the operation dyad A–B would facilitate sequences containing A–B in either the first or second position (e.g., facilitation for both A–B–D and D–A–B). The latter possibility seems to imply a more abstract form of representation for sequence information because actual position of subsequences is lost in the representation.

**Associative chains.** A third possibility is that people learn the order of operations as a complex chain that exceeds simple transition memory. In the case of three operations, A–B–C, they do not simply learn two transitions A–B and B–C; they learn the complete sequence of A–B–C. This differs from composition in that there is no assumption of unitary representation and all-or-none execution. Instead, A is assumed to prime B based on prior associative knowledge. Then, the combination of A–B primes C in the same manner. An associative mechanism such as this has been posited by MacKay (1987) for language acquisition.

### **Independence of sequence and instance information**

Another representation question concerns the degree to which memory for order of operations can be influenced by the surface content of items. These “instance effects” can be potent in some situations. If the two types of information—processing sequence memory and instance memory—are independent, as might be the case in complex associative chains or even simple transition representation, then item surface structure should not interact with sequence effects (i.e., facilitation caused by overlapping sequences of operations). It is unclear whether a composition model could accommodate such independence. Carlson and Schneider (1989; Carlson et al., 1989) argued that composition within a production system model must retain item information to work. Others, such as Anderson (1989) disagreed with this interpretation.

An adaptive system might favor general sequence information under some circumstances (e.g., few sequences of operations but many different item surface structures) and instance-specific information under other circumstances (e.g., few different item surface structures compared with the number of different sequences). Thus the ability to assess independence might vary as a function of the stimulus set under consideration. At any rate, there is little evidence available regarding instance and sequence independence.

In this article we present two experiments aimed at addressing the representation questions raised earlier. In the first experiment, we attempted to determine the level at which a three-rule sequence is learned. Are three-rule sequences learned as dyads or triads, and if they are learned as triads, is composition or associative chain representation more likely? In the second experiment, we examined the effects of instance information on sequence learning. Could similarity at the item level bias how sequence information is retrieved and used? This information sheds light on the relative independence of these two types of memory.

### **EXPERIMENT 1**

---

In Experiment 1 we taught participants a skill that entailed the sequential application of three rules. Three-rule sequences were drawn from a population of all possible orderings of four computational rules. During training, participants saw only a subset of the possible orderings of the rules. During transfer, participants saw all possible orderings. Presentation of rules during training was counterbalanced such that each rule was presented an equal number of times in each serial posi-

tion. This equalized the strengthening of individual rules by serial position.

The question of interest was addressed during transfer. All transfer items were different from training items in terms of their surface structure. Thus instance effects were equalized for old and new sequences in this experiment. Questions of sequence representation were addressed by varying the similarity of rule combinations in transfer to the original training trials. Transfer rule sequences could match training rule sequences in either the first two rules (e.g., A–B of the rule sequence A–B–C; we call this a *first rule dyad match*), the second two rules (e.g., B–C of the rule sequence A–B–C; we call this a *second rule dyad match*), both the first two rules and the second two rules but not the all three rules (the first and second rule dyads match, but not the rule triad; this was possible because a transfer item could match the first rule dyad of one training sequence and the second rule dyad of a different training sequence), or neither the first two rules nor the second two rules (no dyads match). It was also possible to match the rule triad, which implied a match of the first and second rule dyads (these were training rule sequences seen during transfer with new item content). First we consider the predictions of each theory of sequence representation. We made predictions about new transfer trial performance relative to performance on training trials (i.e., triad matches). These predictions are summarized in Figure 1.

A dyad transition model of sequence representation makes the simple prediction that latency and errors will increase for new transfer trials to the extent that dyad transitions differ from those in training sequences. As shown in the left two panels of Figure 1, when both dyads are new, latency and errors will be greatest. When only one dyad is new, latency and errors will be increased to the same extent, regardless of which dyad is new. Of particular importance, latency and errors for trials with two old dyads (but a new triad) should not differ from latency and errors for training trials.

A composition model of sequence representation makes different predictions for both latency and errors than does the dyad transition model. As shown in the upper middle panel of Figure 1, it predicts longer latency when the first dyad differs from training sequences. A new first dyad prevents the composed representation from firing, so it is irrelevant whether the second dyad is old or new. In contrast, an old first dyad is assumed to be sufficient to trigger the composed production, so latency for an old first dyad trial should not differ from that of training sequences that rely on the same representation. With respect to errors, a composition model makes the unique prediction that there will be more errors in old first dyad trials than in new first dyad trials (see



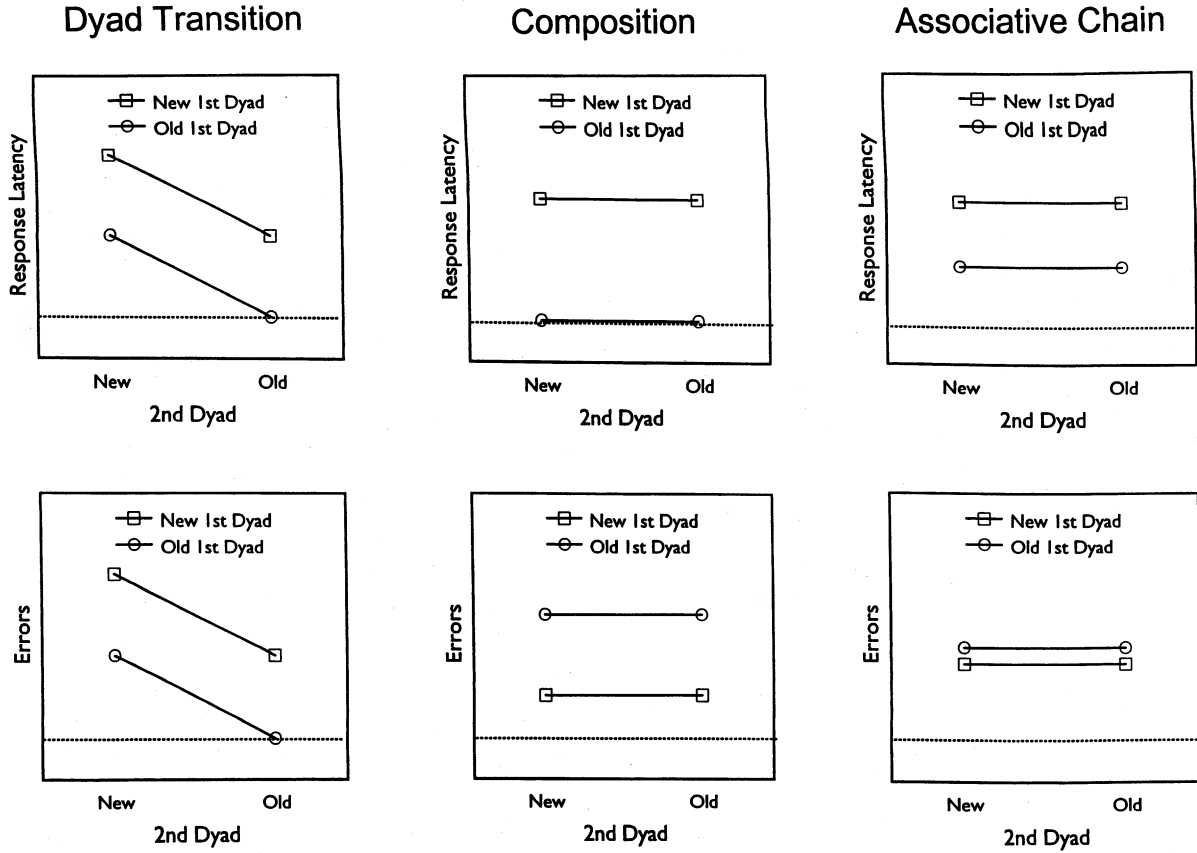


Figure 1. Predicted patterns of response latency and error for different models of sequence knowledge representation

the lower middle panel of Figure 1). As noted earlier, an old first dyad is expected to invoke the all-or-nothing execution of the complete sequence representation. This should produce a high rate of garden path errors. Note that new first dyad trials are also expected to produce more errors than training sequences but not as many errors as old first dyad sequences. The errors associated with new first dyads reflect the lower reliability of reverting to weaker representations (e.g., declarative or initial procedural knowledge for individual rule components).

Finally, an associative chain model of sequence representation makes predictions about transfer performance that are distinct from either dyad transition or composition models (see the right two panels of Figure 1). As with composition, any trial that begins with a new dyad is expected to have the longest latency. In contrast to composition, an associative chain model predicts that new sequences in transfer that begin with old first dyads will produce longer response latency than will training sequences. On these trials, there is partial facilitation from the initial match with the associative chain representation, but latency is increased by the need to revert to other representations to complete the trial (i.e., declarative knowledge or procedural representations of individual component rules). Furthermore, because associative chains are not all-or-nothing in their execution, there is no prediction of high error rates for trials that begin with an old first dyad but end in a new way, as was the case with composition. We would expect some garden path errors on trials that begin with a familiar dyad but end differently. However, there is no reason to expect the frequency of these errors to be higher than errors caused by reverting to weaker representations.

## METHOD

### Experimental task

The skill task used was number reduction, which was a modification of a task originally developed by Thurstone and Thurstone (1941). This task was chosen because it allowed us to train participants to a high level of skill in a short period of time and because it allowed us great control over which rule sequences, and actual instances of these sequences, participants were exposed to. In this version of number reduction, participants were taught a set of four rules for reducing four-digit number strings to a single-digit response. Participants apply the rules to pairs of digits, proceeding from left to right. The application of each rule yields a single-digit answer to the pair of digits. This digit becomes the first digit in the next pair. Processing proceeds until only a single digit remains.

The task is best understood by example. Consider the string "4568." The four rules participants learn are (a) the *same* rule, which states that if two digits are

the same (e.g., 55), the answer is that same digit (5); (b) the *contiguous* rule, which states that if two digits begin an ascending or descending series (e.g., 67 or 43), the answer is the next digit in that series (8 or 2); (c) the *midpoint* rule, which states that if two digits differ by two (e.g., 35), the answer is the digit midway between them (4); and (d) the *last* rule, which states that if two digits differ by more than two (e.g., 38), the answer is the latter of the two digits (8). Our example (which represents the contiguous–same–midpoint rule sequence) would be solved as follows: 45 → 6. Six becomes the first digit in the next pairing, 66 → 6. The final pairing is therefore 68 → 7, and the participant would respond by pressing “7.” Note that the participant does not input the intermediate responses; instead these operations are performed in the participant’s head.

Each four-digit stimulus string therefore requires the application of three rules for its solution. Strings are designed so that no rule appears more than once per string and that over strings (within an individual) the frequency of occurrence of all rules in all serial positions is balanced.

Participants performed the number reduction task on IBM compatible microcomputers with standard keyboards and SVGA monitors. Materials were presented in 24 × 80 text mode. The software was written to achieve millisecond timing of response latency and to record detected and undetected errors (Walker, 1985).

## Participants

Participants were 67 undergraduate students at the University of Utah (32 men, 35 women). All participants were solicited through campus advertisements and were paid \$5 per hour for their participation.

## Procedure

All participants performed the number reduction task for five sessions. The first four of these were training sessions designed to build skill in the task, and the fifth session was a transfer session designed to test our hypotheses.

During training, participants saw items containing 8 of the 24 possible three-rule sequences using each rule once (e.g., same–last–midpoint). The eight sequences were chosen randomly for each participant, and each individual rule was balanced for each serial position (i.e., first, second, or third) across the eight sequences. This meant that each component rule was the initial rule for two sequences. The remaining two rules for each set of sequences with the same initial rule consisted of two different component rules reversed in order across the two sequences. For example, if same–last–midpoint was selected as a training sequence (referred to as “old” sequences during transfer), the other old sequence would be same–midpoint–last.

For each of the eight old sequences, 15 of the possible 24 instances (e.g., “2248” would be an instance of same–midpoint–last) were randomly selected for each person. Each sequence was seen three times per block of trials. So all 15 instances were used across every 5 blocks. During training, participants solved 10 blocks of 24 trials during the first session and 15 blocks of 24 trials during the second, third, and fourth sessions.

The transfer session consisted of 18 blocks of 24 trials. All 24 possible sequences were presented during each block of transfer. One third of these sequences were seen during training (“training trials”), and two thirds of them were new (i.e., not seen during training). However, training sequences never consisted of old items (actual items that were presented during training). New instances were used so that differences between training and new sequences could be attributed solely to differences in sequence memory. Nine instances each were used for training and new sequences during transfer. Each of these instances was seen once during the first nine blocks of transfer and once during the second nine blocks of transfer. The nine instances were randomly determined for each participant. For the training sequences it was the nine instances that were not randomly selected for use in the training sessions. For the new sequences it was a random 9 out of the possible 24 instances per sequence.

Half of the 16 new sequences in the transfer session matched training sequences in their first two rules (denoted “old first dyad” sequences), while the other half did not (denoted “new first dyad” sequences). Half of both the old and new first dyad sequences matched training sequences in the last two rules (denoted “old second dyad” sequences) and half did not (denoted “new second dyad” sequences). Thus, among new sequences, old and new first and second dyads were completely crossed, with four sequences of each possible combination. Table 1 presents a hypothetical participant’s assignment of rules to categories.

**Performance goals and error detection**

During training, participants were encouraged to answer items as quickly as possible while maintaining a 90% accuracy rate. To encourage both speed and

Table 1. Hypothetical example of assignment of rule sequences to categories

| Training sequences |            |       |       |
|--------------------|------------|-------|-------|
| S–M–C              | C–S–L      | M–L–C | L–M–S |
| S–C–M              | C–L–S      | M–C–L | L–S–M |
|                    | First dyad |       |       |
|                    |            | Old   | New   |
|                    | Old        | S–C–L | S–L–C |
|                    |            | C–S–M | C–M–S |
|                    |            | M–L–S | M–S–L |
| Second dyad        |            | L–M–C | L–C–M |
|                    | New        | S–M–L | S–L–M |
|                    |            | C–L–M | C–M–L |
|                    |            | M–C–S | M–S–C |
|                    |            | L–S–C | L–C–S |

*Note.* In transfer, all instances of all sequences were new. S = same rule; C = contiguous rule; M = midpoint rule; L = last rule.

accuracy, the following feedback was provided: Response latency was presented for 1 s after correct responses. Following incorrect responses, the word *WRONG* and a low tone was presented for 2 s. After each block of 24 trials, the overall accuracy (percentage correct) and median latency were presented along with conditional instructions for performance on the next block. If a participant had an error rate of 15% or more, he or she was instructed to slow down to make fewer errors. If a participant had an error rate of less than 5%, he or she was told that they probably were not responding as quickly as they could. In all cases, participants were told that their goal for the next block of trials was to go faster than they did in the previous block and still get about 90% correct. After this instruction, they were shown a summary of median latency for all previous blocks.

During transfer, participants were instructed that their new performance goal was to respond as quickly as possible while achieving 100% rather than 90% accuracy. To make this goal more attainable, they were given an opportunity to correct any mistakes that they made in trying to respond quickly. After a response, participants could press the keyboard spacebar to retake the previous trial. In conjunction with this new goal and procedure, accuracy feedback was no longer provided after incorrect trials or at the end of each block. However, median latency feedback was still provided after each block.

By allowing participants to retake error trials, we were able not only to increase the accuracy requirement during transfer but also to separate those error trials into detected errors (i.e., trials on which the participant pressed the spacebar and retook the trial) and undetected errors (i.e., trials on which the participant made an error but did not retake the trial). This method for determining undetected errors has been successfully used elsewhere (Woltz et al., 2000).

## RESULTS AND DISCUSSION

### Training data

Figure 2a presents the mean latency and error rates during the training blocks of the first four sessions. The first session was 10 blocks long, and the three subsequent training sessions were 15 blocks each. As seen in Figure 2a, participants averaged approximately 10% errors in all four sessions. Also, there was a tendency for participants to be more accurate in the beginning blocks of each session. Mean performance latency showed a steady decline over trial blocks that was well described ( $R^2 = .99$ ) by the power law of practice (Newell & Rosenbloom, 1981).

### Transfer trial latency

We analyzed the latency data for correct responses during the transfer session to test for facilitation caused by first dyad match, second dyad match, and triad match (i.e., training sequences). Figure 2b presents the latency means for transfer trials broken down by trial type. Latency for training sequence trials was shorter than that for new sequence tri-

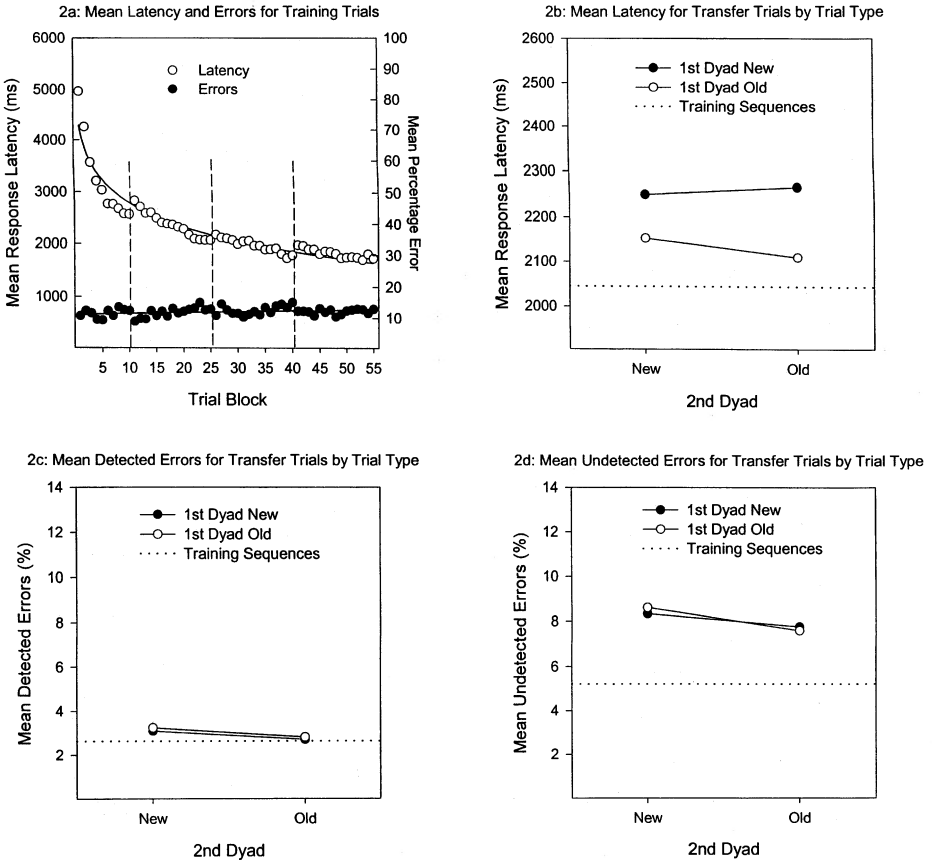


Figure 2. Experiment 1 results

als (collapsed over all trial types),  $F(1, 66) = 63.53$ ,  $MSE = 13,221$ ,  $p < .001$ . This finding supports the general contention that sequence memory exists, and it facilitates performance on matching sequences even when instances are new.

Three planned contrasts in the latency data tested the predictions made by the different sequence representation models. First, the difference between old first dyad trials and new first dyad trials was statistically significant,  $F(1, 66) = 26.84$ ,  $MSE = 42,376$ ,  $p < .001$ , with old first dyad trials being faster by approximately 130 ms. Thus, information about the first two rules and the transition between them is being represented in memory. All three models of sequence memory representation made predictions consistent with this finding.

Second, old second dyad trials did not differ significantly from new second dyad trials,  $F(1, 66) < 1$ . A match in the second dyad did not result in better performance than a nonmatch, and this was true for both old first dyads and new first dyads; the interaction between first dyad and second dyad was not significant,  $F(1, 66) = 1.59$ ,  $MSE = 42,117$ ,  $p > .10$ . This finding was consistent with the composition and associative chain models of sequence memory but not with the dyad transition model that postulated second dyad facilitation regardless of first dyad or triad consistency.

Finally, the difference between training trials and trials with old first and second dyads was statistically significant,  $F(1, 66) = 8.52$ ,  $MSE = 18,401$ ,  $p < .01$ . Training trials were approximately 70 ms faster. The associative chain model was the only model to predict a difference between these trials. The dyad transition model predicts that old first and second dyad trials would be performed as quickly as old triad trials because they both have the same number of old dyads. The composition model assumes that an old first dyad triggers an all-or-nothing execution of a full training sequence, so the response time of any sequence beginning with an old first dyad should be equivalent to that of the training sequences. This finding supports the notion that information about the entire sequence of processing operations in this task is represented in memory (i.e., the triad) and that it is probably represented as a complex associative chain rather than as a unitized representation that executes in total.

### Transfer trial errors

Figures 2c and 2d present the error data broken down by trial type. Figure 2c presents the data for detected errors, and Figure 2d presents the data for undetected errors. Detected and undetected errors were analyzed separately because they are presumed to represent different mechanisms. Undetected errors are assumed to reflect primarily cognitive slips associated with skilled memory representations.

As is evident in Figure 2c, there were few detected errors in this task, and there was little variation by trial type. Training sequence trials did not differ significantly from new trials,  $F(1, 66) = 2.31$ ,  $MSE = 2.55$ ,  $p > .10$ . Furthermore, old first dyad trials did not differ significantly from new first dyad trials,  $F(1, 66) < 1$ , and old second dyad trials did not differ significantly from new second trials,  $F(1, 66) = 1.60$ ,  $MSE = 6.17$ ,  $p > .10$ . The interaction between first dyad and second dyad was also not significant,  $F(1, 66) < 1$ .

For undetected errors (Figure 2d), there were significantly more errors on new trials than on training sequence trials,  $F(1, 66) = 19.16$ ,  $MSE = 14.78$ ,  $p < .001$ . This finding is consistent with previous research

on undetected errors in the number reduction task (Woltz et al., 1996, 2000), and it is consistent with all three models of sequence representation. However, there was no difference in the number of undetected errors between either old and new first dyads,  $F(1, 66) = 0.02$ ,  $MSE = 19.96$ ,  $p > .10$ , or old and new second dyads,  $F(1, 66) = 1.56$ ,  $MSE = 25.96$ ,  $p > .10$ . Also, the interaction between first and second dyad was not significant,  $F(1, 66) = 0.57$ ,  $MSE = 13.80$ ,  $p > .10$ . These findings are consistent with the associative chain model and inconsistent with the dyad transition and composition models (see lower panels of Figure 1). The composition model predicts an effect for the first dyad match, with more errors on old first dyad trials. The dyad transition model predicts an effect for both first dyad match (i.e., more errors in new first dyad trials) and second dyad match (i.e., more errors in new second dyad trials). Neither effect approached statistical significance; however, the data were quite consistent with the predictions of the associative chain model.

### Conclusions

Three models depicting how processing sequence information is represented in memory made contrasting predictions for skill transfer performance. Comparisons between different transfer conditions in both latency and error data were inconsistent with predictions of the dyad transition and composition models of sequence representation but were consistent with predictions from an associative chain model. The findings suggest that the acquisition of complex sequence information partly underlies performance improvements in a multistep cognitive skill. Furthermore, the sequence information probably is represented as associated links between the entire sequence of operations. The associative links presumably enhance performance by priming subsequent operations. The string of associations does not appear to be triggered in an all-or-nothing fashion, as would be expected if sequence knowledge were represented as a unitized whole (e.g., composed productions).

## EXPERIMENT 2

---

The pattern of latency and error data in the various transfer conditions of Experiment 1 led us to reject composition and dyad transition models of sequence representation in favor of a complex associative chain model. However, the design of the experimental task may have unduly disadvantaged the composition model. Composition might be more likely to occur in skills that take fewer sequences to be learned.



Also, composed sequences might be triggered during transfer in the manner predicted for Experiment 1 only when there is a close match between training sequence surface structure and transfer sequence surface structure. In Experiment 1, digit strings presented during transfer were always different from those presented during training, even when the sequence of operations was identical to those from training.

The evidence thus far suggests that sequence memory has a degree of generality. That is, memory for the sequence of processing operations facilitates performance even with new surface structure of individual trials (i.e., new data on which the sequence of operations executes). In Experiment 1, we contrasted composition and other models under conditions that assumed data-general sequence representation. However, it is not clear that the composition mechanism is capable of handling such generality. Carlson and Schneider (1989; Carlson et al., 1989) argued that for a composition mechanism to work, it logically must incorporate data-specific aspects of the particular instance viewed. In tasks such as number reduction, the output of one step determines the input to the subsequent step (i.e., cascaded task component steps). Furthermore, intermediate step solutions determine which subsequent operations are applicable. Real-time processing adaptations that depend on intermediate solutions are inconsistent with the notion of all-or-nothing execution of a composed set of steps. Under this view, composition should not be possible unless instances were consistent in both training and transfer.

Anderson (1989) disagreed with the need to retain item surface structure within composed productions. He allowed variables to be composed in place of specific intermediate results, thus allowing for instance-independent sequence memory. Although the finding of instance-general sequence memory effects seems to support Anderson's position, the data from Experiment 1 were otherwise inconsistent with a composition explanation. It should also be noted that Anderson (1993) dropped the composition mechanism in a later version of the ACT theory.

Experiment 2 was designed primarily to assess whether transfer performance data conform to general predictions of the composition model when new sequence transfer trials resembled training trials in the first dyad and in the first three digits. The composition model predicts that when new sequences begin like training sequences in the first dyad and when they are identical to a training instance that had been repeatedly practiced, latency will be as fast as that for old training instances and undetected errors will be substantially higher than in any other trial condition. In addition, if all-or-nothing execution of composed productions is triggered by this "partial match" of training stimulus conditions,

the latency of undetected errors should not differ from the latency on training sequences performed correctly.

## METHOD

### Experimental task

The experimental skill acquisition task was number reduction, which was described earlier.

### Participants

Participants were 51 University of Utah students (27 men, 24 women). Participants were paid \$5 per hour for their participation.

### Procedure

All participants performed the number reduction task in three sessions, with 18 blocks per session. The first two of these were training sessions, designed to develop skill, as were the first six blocks of session three (this was different than in Experiment 1). Transfer consisted of the final 12 blocks of the third session. The continuation of training at the beginning of the final session made the transition to transfer less apparent to participants and thus increased the likelihood that they would rely on their skilled memories to guide their performance.

During training participants practiced four rule sequences, with each sequence being represented by 12 instances per sequence. During transfer, participants received a total of eight rule sequences, with each rule sequence being represented by 12 instances per sequence. Every two blocks of transfer trials represented a complete replication of the design.

Of the eight transfer sequences, four were old (i.e., seen during training) and four were new. The four old sequences were represented by two categories of instances: old instances seen during training (designated "old/old"), and new instances (designated "old/new"). The four new sequences matched the old sequences in the first rule dyad (A-B) and were also represented by two categories of instances: instances that matched old instances in the first three digits (e.g., "4656," which matches the old instance "4659" in the first three digits, although these represent different sequences; these instances were designated "new/old") and instances that did not match old instances (designated "new/new").

Although the labels for our trial conditions give the appearance of a  $2 \times 2$  crossed design, this was not really the case. The new/old condition, which was a new sequence with an "old" instance, matched training instances only in the first three of the four digits. By virtue of it being a new sequence, it was not possible for it to match a training instance in all four digits. Thus this condition is not completely comparable to the old/old condition, in which the old instance matched a training instance in all four digits. Because of this difference in the meaning of "old instance" across old and new sequences, the data were not analyzed in a traditional crossed analysis of variance design.

Which four sequences were used as training and which four were used as new sequences during transfer was counterbalanced over participants. This allowed us to measure whether our effects were strongly determined by the particular rule sequences and instances used.

As in Experiment 1, during the final session, participants' performance goal was changed from 90% accuracy at maximum speed to 100% accuracy at maximum speed. Participants were able to press the spacebar to retake any trial on which they thought they had made an error. This allowed us to separate error trials into detected and undetected errors.

## RESULTS AND DISCUSSION

### Training data

As in Experiment 1, we examined the training data to ensure that participants attained a high level of skill on the number reduction task. Figure 3a presents the mean latency and error rates during the training blocks of the first three sessions. The first two sessions were 18 blocks long; the first six blocks of the third session also served as training. As seen in Figure 3a, participants averaged approximately 10% errors in during the first two sessions. However, at the beginning of the third session participants were instructed to attempt to achieve 100% accuracy. During the training blocks in this session, participants' error rates dropped substantially, and latencies increased. For the first two sessions, mean performance latency showed a steady decline over trial blocks that was well described ( $R^2 = .98$ ) by the power law of practice (Newell & Rosenbloom, 1981).

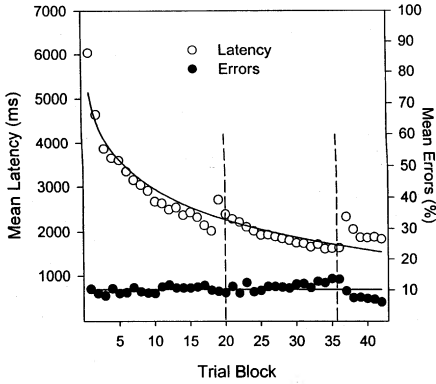
### Transfer trial latency

We analyzed the latency data for correct responses during the transfer blocks of the third session to test for performance differences between trial types. Because the design was replicated over every set of two blocks, data were collapsed over pairs of blocks, yielding six block pairs. Figure 3b presents the latency means for transfer trials broken down by trial type.

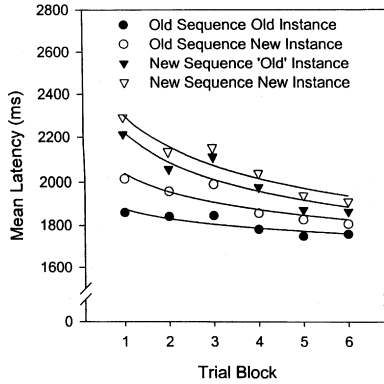
Two trial type contrasts were of general interest. First, the contrast of old sequence, new instance versus new sequence, new instance tested the presence of data-general sequence memory. This contrast was statistically significant,  $F(1, 49) = 42.42$ ,  $MSE = 5,087,989$ ,  $p < .001$ , with old sequences being approximately 170 ms faster than new sequences. As in Experiment 1, there was strong support for facilitation caused by the same operations being applied in the same order, even though the data being operated on were new.

Second, the contrast of old sequence, old instance versus old sequence, new instance tested the role of instance memory beyond that

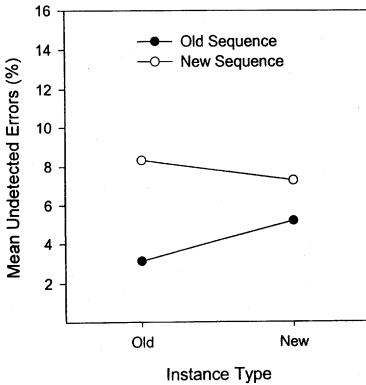
3a: Mean Latency and Errors for Training Trials



3b: Mean Latency for Transfer Trials by Type



3c: Mean Undetected Transfer Error by Trial Type



3d. Median Latency for Correct and Error Responses by Type

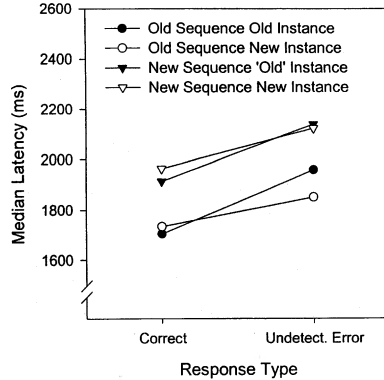


Figure 3. Experiment 2 results

of sequence memory, that is, facilitation caused by identical item content or surface structure in training sequence trials. This contrast was also statistically significant,  $F(1, 49) = 34.39$ ,  $MSE = 47,518$ ,  $p < .001$ , with old instances being approximately 105 ms faster than new instances. Clearly, some portion of participants' performance on training sequences was instance based. This was consistent with previous research using the current task paradigm (Woltz et al., 1996, 2000) and research using other tasks (Carlson & Lundy, 1992; Logan, 1988).

**Transfer trial errors**

Transfer trial errors were separated into detected errors and undetected errors and analyzed in an analogous way to the latency data.

However, the detected error rate was low and comparable across trial conditions, similar to what was found in Experiment 1 (2–3% detected errors). Consequently, we report only data for undetected errors here. In addition, because the undetected error distributions were skewed, a nonparametric Wilcoxon signed ranks test was used rather than analysis of variance.

Figure 3c presents median undetected errors as a function of trial type. Undetected error rates ranged between 3.0% and 8.5% across trial types. The measure of sequence memory, the contrast of old sequence, new instance versus new sequence, new instance, was statistically significant, Wilcoxon  $Z = 3.133$ ,  $p < .01$ . So there was evidence of processing sequence facilitation in the undetected error data, as there was in the latency data.

The measure of instance-based facilitation, the contrast of old sequence, old instance versus old sequence, new instance, was also statistically significant, Wilcoxon  $Z = 3.727$ ,  $p < .001$ . Again, as in the latency data, performance was to some degree instance based.

Composition made the unique prediction that a partial match of the instance stem (the first three digits) in new sequences would cause the firing of an incorrect an “old” composed production developed during training. This would result in a higher undetected error rate in the new sequence, old instance condition than in the new sequence, new instance condition. Furthermore, these undetected errors should have latencies that are equivalent to correct responses in the old sequence, old instance condition.

The new sequence, old instance versus new sequence, new instance contrast for undetected errors was not statistically significant, Wilcoxon  $Z = 0.675$ ,  $p > .10$ . In addition, we considered the absolute number of errors made in the new sequence, old instance condition. The error rate here was 8.33%, which was at best moderate. If this condition represented the firing of composed productions caused by a partial match of the enabling conditions, we would have expected a much higher error rate. These data seem more consistent with an associative chain representation of sequence information.

Figure 3d presents the latency data for undetected errors and correct responses in Experiment 2 as a function of item category. The number of observations per condition is 28 rather than 51 because some participants made no undetected errors in some conditions. As can be seen in the figure, there is a difference in latency between undetected errors in the new sequence, old instance condition ( $Mdn = 2,136$  ms) and correct responses in the old sequence, old instance condition ( $Mdn = 1,704$  ms). A test of the contrast was statistically significant, Wilcoxon  $Z = 4.099$ ,  $p < .001$ . Thus, analysis of both the undetected er-

ror data and the latency data failed to support the predictions of the composition model.

### **Conclusions**

The results of Experiment 2, though inconsistent with a composition representation of sequence information, were consistent with an associative chain representation of sequence memory. There was clear support for both sequence-based and instance-based memory effects in the latency and undetected error data. Both composition and associate chain representations predict such effects. However, additional predictions made by the composition model (i.e., all-or-none firing of composed productions, triggered by a partial match of the production's enabling conditions) were not supported in either the undetected error data or the latency data for these errors. Thus, as in Experiment 1, the data are more consistent with an associative chain representation.

## **GENERAL DISCUSSION**

---

Learning new cognitive skills often requires that we learn how to order frequently used component operations to solve the problem at hand. The evidence we have presented here and elsewhere (Bell, Gardner, & Woltz, 1997; Woltz et al., 1996, 2000) supports an important role for an abstract memory for the sequence of operations that is not tied to the actual instances encountered. This memory appears to be implicit in nature; participants in our studies do not appear to be consciously aware that some sequences of mental operations have been frequently seen and others have not (Woltz et al., 2000). Nonetheless, this memory is revealed in decreased latencies and lower error rates on items containing operation orders previously encountered. Similar evidence has been reported for participants learning sequences of responses in serial choice reaction time tasks (Cleeremans & McClelland, 1991); however, in these tasks the operation and the data operated on are perfectly correlated (i.e., a particular stimulus requires the pressing of a particular computer key every time).

In this article we have explored how such sequence memory is represented. Our results supported an associative chain representation of sequence memory. Such a representation is consistent with models of sequential processing in the literature such as MacKay's (1987). Our results were inconsistent with models based only on representation of rule dyads or transitions between rule pairs because there was a significant latency effect for a match of rule triads. Our results were also inconsistent with a composition model. Such a model hypothesizes that

practice results in sets of rules being restructured into single unitized wholes that fire in an all-or-none fashion. Our evidence suggests that processing can be interrupted when a late mismatch occurs. Undetected error rates and undetected error latencies did not match the prediction that a partial match of composed productions' enabling conditions would lead to performance that is fast but error prone. Of course, composition models could be amended in ways that make predictions that are consistent with our data. But we wonder whether such amended composition models could be differentiated from associative chain models. Without differentiating predictions, the differences would be only in the descriptive language used to present the models.

Other models have been proposed that might also account for our data. Cleeremans and McClelland (1991) successfully modeled sequence learning in a serial choice reaction time task using a simple recurrent network (Cleeremans, Servan-Schreiber, & McClelland, 1989) within a parallel distributed processing framework. Although the current study was not designed to test such a model, it is certainly possible that other models can be devised to account for the current data if appropriate processing assumptions are made to complement the representational assumptions.

Our data support the findings of Carlson and Lundy (1992), who found that consistent data were necessary for composition to occur. Despite having to learn only four sequences during training, our data from Experiment 2 did not support the development of composition. We also note that the details of a composition model are far from clear. In particular, Carlson's (Carlson & Schneider, 1989; Carlson et al., 1989) logical argument about the difficulty of composing a production without also composing the actual data remains. Without data consistency, it appears that composition is difficult, if not impossible, to achieve.

## Notes

Research reported in this article was supported in part by a grant from the U.S. Air Force Office of Scientific Research (Grant F49620-93-0094) to Woltz and Gardner.

Correspondence about this article should be addressed to Michael K. Gardner, 327 MBH, Department of Educational Psychology, University of Utah, Salt Lake City, UT 84112 (e-mail: gardner@ed.utah.edu). Received for publication March 30, 2000; revision received February 14, 2001.

## References

- Ackerman, P. L. (1988). Determinants of individual differences during skill acquisition: Cognitive abilities and information processing. *Journal of Experimental Psychology: General*, *117*, 288-318.

- Anderson, J. R. (1983). *The architecture of cognition*. Cambridge, MA: Harvard University Press.
- Anderson, J. R. (1987). Skill acquisition: Compilation of weak-method problem solutions. *Psychological Review*, *94*, 192–210.
- Anderson, J. R. (1989). Practice, working memory, and the ACT\* theory of skill acquisition: A comment on Carlson, Sullivan, and Schneider (1989). *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 527–530.
- Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Erlbaum.
- Bell, B. G., Gardner, M. K., & Woltz, D. J. (1997). Individual differences in skilled performance. *Learning and Individual Differences*, *9*(1), 43–61.
- Carlson, R. A., & Lundy, D. H. (1992). Consistency and restructuring in learning cognitive procedural sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 517–526.
- Carlson, R. A., & Schneider, W. (1989). Practice effects and composition: A reply to Anderson. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 531–533.
- Carlson, R. A., Sullivan, M. A., & Schneider, W. (1989). Practice and working memory effects in building procedural skill. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *15*, 517–526.
- Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, *120*, 235–253.
- Cleeremans, A., Servan-Schreiber, D., & McClelland, J. L. (1989). Finite state automata and simple recurrent networks. *Neural Computation*, *1*, 372–381.
- Cohen, A., Ivry, R. I., & Keele, S. W. (1990). Attention and structure in sequence learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*, 17–30.
- Frensch, P. A. (1991). Transfer of composed knowledge in a multistep serial task. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *17*, 997–1016.
- Koch, I., & Hoffmann, J. (2000). The role of stimulus-based and response-based spatial information in sequence learning. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*(4), 863–882.
- LaBerge, D., & Samuels, S. J. (1974). Toward a theory of automatic information processing in reading. *Cognitive Psychology*, *6*, 293–323.
- Logan, G. D. (1978). Attention in character classification: Evidence for automaticity of component stages. *Journal of Experimental Psychology: General*, *107*, 32–63.
- Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492–527.
- Luchins, A. S. (1942). Mechanization in problem solving. *Psychological Monographs*, *54* (Whole No. 248).
- Lundy, D. H., Wegner, J. L., Schmidt, R. J., & Carlson, R. A. (1994). Serial step learning of cognitive sequences. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 1183–1195.
- MacKay, D. G. (1982). The problems of flexibility, fluency, and speed–accuracy trade-off in skilled behavior. *Psychological Review*, *89*, 483–506.
- MacKay, D. G. (1987). *The organization of perception and action*. New York: Springer-Verlag.



- McKendree, J. E., & Anderson, J. R. (1987). Frequency and practice effects on the composition of knowledge in LISP evaluation. In J. M. Carroll (Ed.), *Cognitive aspects of human-computer interaction* (pp. 236-259). Cambridge, MA: MIT Press.
- Newell, A., & Rosenbloom, P. S. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.), *Cognitive skills and their acquisition*. Hillsdale, NJ: Erlbaum.
- Nissen, M. J., & Bullemer, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, *19*, 1-32.
- Schneider, W. (1985). Toward a model of attention and the development of automaticity. In M. Posner & O. S. Marin (Eds.), *Attention and performance XI* (pp. 475-492). Hillsdale, NJ: Erlbaum.
- Schneider, W., & Shiffrin, R. M. (1977). Controlled and automatic human information processing: I. Detection, search, and attention. *Psychological Review*, *84*, 1-66.
- Shiffrin, R. M., & Schneider, W. (1977). Controlled and automatic human information processing: II. Perceptual learning, automatic attending, and a general theory. *Psychological Review*, *84*, 127-190.
- Stadler, M. A., & Neely, C. B. (1997). Effects of sequence length and structure on implicit serial learning. *Psychological Research*, *60*, 14-23.
- Thurstone, L. L., & Thurstone, T. G. (1941). *Factorial studies of intelligence*. Chicago: University of Chicago Press.
- Walker, R. (1985). *PLATS: Software for cognitive tasks*. Unpublished computer program.
- Wegner, J. L., & Carson, R. A. (1996). Cognitive sequence knowledge: What is learned? *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 599-619.
- Woltz, D. J., Bell, B. G., Kyllonen, P. C., & Gardner, M. K. (1996). Memory for order of operations in the acquisition and transfer of sequential cognitive skills. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*, 438-457.
- Woltz, D. J., Gardner, M. K., & Bell, B. G. (2000). Negative transfer errors in sequential cognitive skills: Strong-but-wrong sequence application. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *26*, 601-625.