Kognitionswissenschaft (1999) 8: 5–19

Kognitions-
wissenschaft
© Springer-Verlag 1999

The dynamics of cognition: An ACT-R model of cognitive arithmetic

Christian Lebiere

Psychology Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA
(e-mail: cl+@cmu.edu; web: http://act.psy.cmu.edu)

Dynamik der Kognition: Ein ACT-R Modell zur Kognitiven Arithmetik

Zusammenfassung. Forschungsarbeiten zur Kognitiven Arithmetik untersuchen die mentale Repräsentation von Zahlen und arithmetischen Fakten sowie die kognitiven Prozesse die diese generieren, abrufen und manipulieren. Das Spannungsfeld zwischen der scheinbar einfachen formalen Struktur dieses Aufgabenbereichs und den Schwierigkeiten, die Kinder bei seiner Bewältigung haben, stellt einen einzigartigen Zugang zum Studium kognitiver Prozesse dar. Der vorliegende Beitrag präsentiert einen Erklärungsansatz der zentralen Befunde des Forschungsgebietes auf der Grundlage eines ACT-R Modells zur Lebenszeit-Simulation des Erwerbs arithmetischen Wissens. Die Anwendung der Bayesischen Lernmechanismen der ACT-R Architektur zeigen auf, wie sich diese Befunde auf die statistische Struktur des Aufgabengebiets zurückführen lassen. Aus den präzisen Vorhersagen der Simulation werden sowohl Hinweise zur Vermittlung arithmetischen Wissens abgeleitet als auch Erkenntnisse über die Architektur ACT-R selbst gewonnen. Im Rahmen einer formalen Analyse wird gezeigt, daß sich die vorgestellte Simulation als dynamisches System betrachten läßt, dessen Lernergebnis unmittelbar von Parametern der Architektur abhängt. Eine Untersuchung der Sensitivität der Parameter der Simulation belegt, daß die Werte, die zur besten Anpassung an die empirischen Daten führen, auch eine in einer optimalen Performanz resultieren. Die Implikationen dieses Ergebnis für die grundlegende Adaptivität menschlicher Kognition werden diskutiert.

Abstract. Cognitive arithmetic studies the mental representation of numbers and arithmetic facts and the processes that create, access, and manipulate them. The contradiction between the apparent straightforwardness of its exact formal structure and the difficulties that every child faces in mastering it provides an important window into human cognition. An ACT-R model is proposed which accounts for the central results of the field through a single simulation of a lifetime of arithmetic learning. The use of the architecture's Bayesian learning mechanisms explains how these effects arise from the statistics of the task. Because of the precise predictions of the simulation, a number of lessons are derived concerning the teaching of arithmetic and the ACT-R architecture itself. A formal analysis establishes that the simulation can be viewed as a dynamical system whose ultimate learning outcome is fundamentally dependent upon some architectural parameters. Finally, an empirical study of the sensitivity of the simulation to its parameters determines that the values that yield the best fit to the data also provide optimal performance. The implications of these findings for the fundamental adaptivity of human cognition are discussed.

1 Introduction

Cognitive arithmetic studies the mental representation of numbers and arithmetic facts (counting, addition, subtraction, multiplication, division) and the processes that create, access, and manipulate them. Arithmetic is one of the fundamental cognitive tasks which humans have to master. Children go through years of formal schooling to learn first the numbers, and then the facts and skills needed to manipulate them. Many adults have not and will never completely master the domain. And yet, it is a task that is trivial for computers to perform correctly. Some tasks, such as chess, are hard for both humans and machines to perform and require years of learning or engineering. Other tasks, such as vision, which seem to come naturally to humans, require much programming for computers to perform even poorly. One can attribute that to humans possessing a complex vision system which resulted from millions of years of evolution but will require painstaking work to reverse-engineer and replicate in computers. But a task such as arithmetic seems so straightforward and easy to accomplish that it is surprising that it takes years of learning for humans to master. This suggests that human cognition at the most basic level embodies some assumptions about its environment that are at odds with the structure of arithmetic as it is taught. Arithmetic, being a formal mathematical theory, assumes a set of precise and immutable objects (numbers), facts, and procedures. Human cognition, on the other hand, has evolved to deal with approximate concepts, a changing environment, and adaptive procedures. Studying how such a flexible system deals with a formal task such as arithmetic provides an excellent window to its assumptions and mechanisms.

ACT-R is a hybrid production-system theory of human cognition (Anderson 1993; Anderson and Lebiere 1998). At the symbolic level, ACT-R is a fairly standard goal-directed production system, with a declarative memory of long-term facts, known as chunks, and a procedural memory holding general production rules. At that level, cognitive arithmetic is a trivial task for ACT-R. All one needs to do is give ACT-R the correct chunks representing arithmetic facts and productions encoding procedures to manipulate them and perfect performance will result. This, however, would not be a very satisfactory model of human, especially children's, performance and ignores the impact of ACT-R's sub-symbolic level. ACT-R is also an activation-based system in which the performance at the symbolic level is controlled by real-valued quantities associated with each symbolic structure. Those quantities are learned according to Bayesian principles to reflect the architecture's environment. Retrieval and matching of memory chunks by production rules is a noisy, approximate process driven by activation rather than the exact matching of conditions. Thus the behavior of the system becomes adaptive, stochastic, and error-prone, matching human behavior better but making cognitive arithmetic a more challenging, but also more interesting, task. Cognitive arithmetic is a task that is both well-suited and challenging to ACT-R for a number of reasons. Unlike tasks artificially designed for the purpose of isolating a particular cognitive mechanism, the learning and performance of arithmetic involves most mechanisms of the architecture. It is therefore an excellent test of whether these parts can perform together as well as separately. Unlike laboratory tasks, large amounts of data are available for every cross-section of the population and every aspect of the task, making it easier to establish the trends being analyzed.

While arithmetical concepts can refer to concrete objects and procedures (e.g., children learn the concept “3” by being shown three rabbits and subtract 2 from 3 by eating two out of three cupcakes), arithmetic also has a fundamentally abstract structure. It is much less likely that people have brain structures optimized for performing arithmetic than, for example, vision or language, and suggests a near-complete reliance on general-purpose learning mechanisms.[1] Since each skill builds on the previous ones, e.g., counting can be used to perform addition which in turn can be used to perform multiplication, learning can thus be a mostly self-contained process, rather than entirely dependent upon external factors such as teaching. Of course, the learning of arithmetic also includes an element of pure memorization, e.g., the multiplication table, and the model will therefore include a mixture of rehearsal and computation. Arithmetic also has an inherently clear, simple and regular structure, with a systematic organization of knowledge into tables of immutable facts. This strong regularity, unlike the many exceptions of tasks such as natural language processing, also helps reduce degrees of freedom in modeling the task and provides a good test of ACT-R's statistical learning. These factors lead to a simpler, more regular model that is more predictive than one with many unanalyzed degrees of freedom. This paper presents an ACT-R model that can account for the main effects of cognitive arithmetic in a single simulation by exploiting the architecture's learning mechanisms.

2 Data

There are two main classes of empirical phenomena in the domain of cognitive arithmetic. One concerns the fact that children, and to a certain degree adults, approach answering arithmetic problems with two basic strategies. One strategy is to simply retrieve the answer. The second strategy, referred to hereafter as the backup strategy or backup computation, is to compute the answer. For example, given a problem such as 3 + 4 children may choose to count (perhaps 4, 5, 6, 7) to provide the answer and given $3 * 4$ they may choose to add to get the answer (perhaps 4 + 4 + 4). This class of empirical phenomena involves how people choose between the computation strategy and the retrieval strategy.

The second class of empirical phenomena involves the problem-size effect.[2] Children and adults take longer to answer problems involving larger numbers and they also make more errors on these problems. In the case of backup computation the reason for this is fairly obvious – one has to count more to add large numbers and one has to add more things when multiplying by a larger number. However, while much reduced, the problem-size effect also occurs for adults. It has been suggested that this is due to residual use of the backup strategy (LeFevre et al. 1996), although recent research put those results in doubt (Kirk and Ashcraft 1997). However, a more fundamental argument is that smaller problems also occur more often, offering greater practice and thus better performance. This unequal problem distribution appears in studies of textbooks (Ashcraft 1987; Ashcraft and Christy 1995; Hamman and Ashcraft 1986; Siegler 1988) but also in the world at large, as many (Benford 1938; Newcomb, 1888; Raimi 1976) have noted.

Ashcraft (1987) reports the change in response time for addition problems in adults, differentiating between tie problems (equal operands), problems involving zero and other non-tie non-zero problems (Fig. 1). While most problems exhibit an increase in response time roughly corresponding to the square of the sum of their operands, the slope for problems involving a zero operand is approximately flat, and the increase in response time for tie problems is much smaller than for other problems. The effect therefore reflects a more complex measure of problem difficulty than simply problem size.

Siegler and Shrager (1984) present the pattern of retrieval errors by four-year-olds for addition facts ranging from 1+1 to 5+5 (Fig. 2). The main effect, similar to the problem-size effect, is an increase in errors for larger facts, with the increase being slightly larger for the addend (second operand) than for the augend (first operand).

Siegler (1988) presents the percentage of errors in the computation of multiplication problems by repeated addition for fourth-graders (Fig. 3). Analogous to the addition

[1] See (Peterson and Simon, in press) for an ACT-R account of subitizing, a skill related to both vision and arithmetic.

[2] We use the term “problem size” here to refer to the size of the numbers involved but other definitions can also be used.

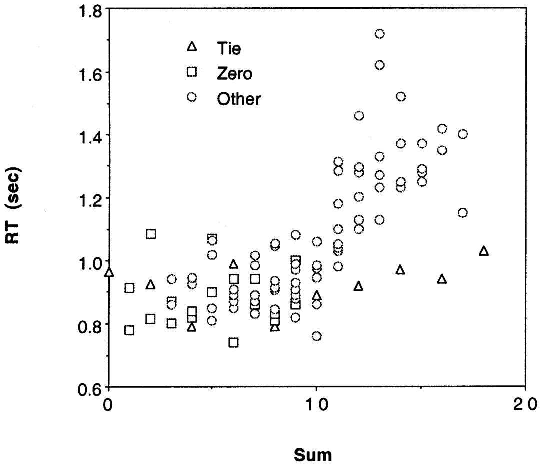

Fig. 1. Problem-size effect for adults

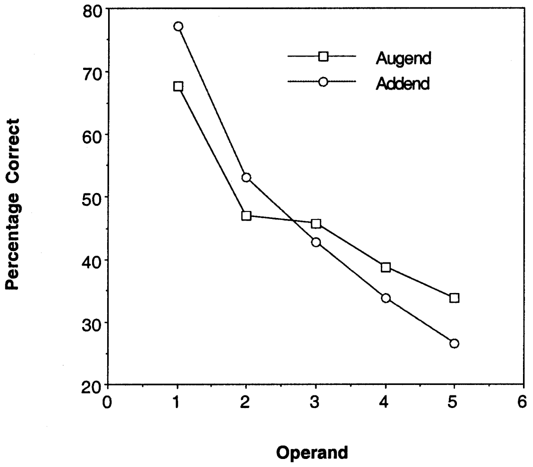

Fig. 2. Percentage of correct addition retrievals per operand

problems, the probability of error increases with the size of both the multiplicand and the multiplier. Particularly remarkable is the very low percentage of errors for the repeated addition of 5.

Finally, Ashcraft (1987) describes the decrease in response time to addition problems across grades, as well as

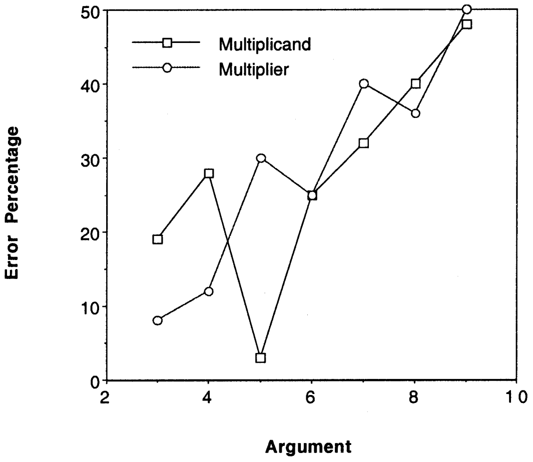

Fig. 3. Percentage of erroneous multiplication computations per operand

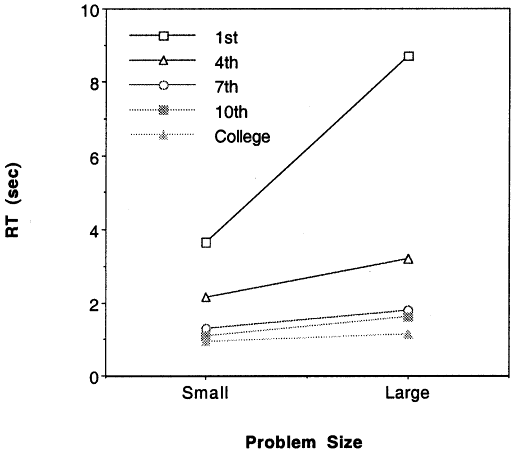

Fig. 4. Problem-size effect across time (grades)

the gradual flattening of the problem-size effect for large vs. small problems (meaning two-digit sum vs. single-digit sum) from first grade to college (Fig. 4).

These effects of problem size and strategy are ubiquitous throughout the literature on cognitive arithmetic (for reviews of the field, see, for example, Ashcraft 1992, 1995; Campbell 1995; Geary 1996). While these effects are by far not the only ones to account for, they constitute a good basis for a comprehensive model of cognitive arithmetic.

3 ACT-R

ACT-R is an activation-based goal-directed production system theory (Anderson 1993; Anderson and Lebiere 1998). Knowledge in ACT-R is divided into declarative knowledge stored in chunks (e.g., arithmetic facts) and procedural knowledge stored in productions (e.g., how to compute the answer to an addition problem). Subsymbolic activation processes control which productions are used and which chunks are retrieved. The parameters of these numerical processes reflect the previous statistics of use for the knowledge structures to which they are attached. They are learned by Bayesian learning mechanisms derived from the rational analysis of cognition (Anderson 1990). The equations governing this subsymbolic level will be described below. The reader should consult Anderson and Lebiere (1998) for additional details of the ACT-R theory.

In ACT-R, the activation of a declarative memory element, or chunk, can be interpreted as reflecting the log posterior odds that the chunk is relevant in the current context. The activation of a chunk is computed as the sum of the base-level activation of that chunk plus the sum for all context elements of their weights (also known as activation source level) times the strength of association between the context element and the chunk. In Bayesian terms, the base-level activation represents the log prior odds of the chunk being relevant and the strength of association represents the log likelihood ratio that the chunk is relevant given the context element. Formally, the activation A_i of chunk i is defined as:

$$A_i = B_i + \sum_j W_j S_{ji} \qquad \textbf{Activation Equation}$$

where B_i is the base-level activation of i, W_j is the attentional weight given the context element j, and S_{ji} is the strength of association from element j to chunk i. An element j is in the focus, or in context, if it is a part of the current goal, and the total amount of attention is divided evenly among goal elements.

The base-level activation of a chunk can learn to reflect the past history of the use of that chunk:

$$B_i = \ln \sum_{j=1}^{n} t_j^{-d} \qquad \textbf{Base-Level Learning Equation}$$

where n is the total number of references to the chunk, t_j is the time elapsed since the jth reference (retrieval or creation) of chunk i, and d is the memory decay rate. With the assumption that occurrences are evenly distributed, the previous equation reduces to a simpler form that is more analytically tractable and can be more efficiently computed:

$$B_i = \ln \frac{n \cdot L^{-d}}{1-d} \qquad \textbf{Optimized Learning Equation}$$

where L is the lifetime of the chunk, i.e., the time since its creation. Similarly, the strengths of associations can learn to reflect the past history of the use of a chunk given its context:

$$S_{ji} = \ln \frac{a \cdot R_{ji}^{*} + F(C_j) \cdot E_{ji}}{a + F(C_j)}$$

Posterior Strength Equation

where R_{ji}^{*} is the prior strength of association, a is the weight given to that prior, $F(C_j)$ is the frequency of j being in the context (i.e., a source of activation in the goal), and E_{ji} is the empirical strength of association. Initially, the strength of association is equal to the prior:

$$\ln(R_{ji}^{*}) = \ln(m/n) \qquad \textbf{Prior Strength Equation}$$

where m is the total number of chunks in declarative memory and n is the number of chunks which contain the source chunk j. Their ratio is a static estimation of the increased likelihood of retrieving chunk i containing the chunk j when j is a source of activation. With extensive experience, the prior is discounted and the strength of association converges to the empirical estimation:

$$\ln(E_{ji}) = \ln \frac{F(N_i \& C_j) \cdot F}{F(N_i) \cdot F(C_j)} \qquad \textbf{Empirical Ratio Equation}$$

where $F(N_i\&C_j)$ is the frequency of chunk i being needed (retrieved) with chunk j in context, $F(N_i)$ is the frequency of i being needed, $F(C_j)$ is the frequency of j being in the context (i.e., a source of activation in the goal), and F is the total number of opportunities (productions matched) since i was created.

In exact matching mode, ACT-R only considers the chunks that match perfectly to the production condition(s). In partial matching mode, every chunk of the correct type is considered, but a mismatch to the production condition results in a penalty being subtracted from the chunk activation to yield its match score:

$$M_{ip} = A_i - MP \cdot \sum_{conditions} (1 - Sim(v,d))$$

Match Equation

where MP is the mismatch penalty constant and $Sim(v,d)$ is the similarity between the desired slot value d specified in the production condition and the actual slot value v contained in the chunk. Gaussian noise of mean 0 and standard deviation σ is also added to the activation and the chunk with the highest final match score is then selected, assuming that it reaches the retrieval threshold τ. If one approximates the Gaussian noise with a logistic distribution, the probability that the match score of chunk i to production p is above the retrieval threshold is:

$$P = \frac{1}{1 + \mathbf{e}^{-\frac{M_{ip}-\tau}{s}}} \qquad \textbf{Retrieval Probability Equation}$$

where $s = \sqrt{3}\sigma/\tau$. If no chunk reaches the retrieval threshold, then a retrieval failure occurs and the next available production is selected. If more than one chunk is competing for retrieval, the probability of chunk i being the one that is retrieved follows the Boltzmann distribution (e.g., Moussouris 1974):

$$P(i) = \frac{\mathbf{e}^{M_{ip}/t}}{\sum_j \mathbf{e}^{M_{jp}/t}} \qquad \textbf{Chunk Choice Equation}$$

where $t = \sqrt{2}s$. The latency T_{ip} to retrieve (match) a chunk i with production p is an exponentially decreasing function of the chunk's match score:

$$Time_{ip} = F\mathbf{e}^{-fM_{ip}} \qquad \textbf{Retrieval Time Equation}$$

where F is a time scaling constant and f an activation scaling constant usually left at its default value of 1. The productions that can apply to the current goal are matched sequentially in decreasing order of expected gain E:

$$E = PG - C \qquad \textbf{Expected Gain Equation}$$

where G is the value of the current goal, P is its probability of success and C is the cost of execution of this and following productions until completion of the goal. Noise is also added to the expected gain value, and thus production selection will also be stochastic and follow the Boltzmann distribution. The probability and cost parameters can also be learned according to the record of success and failure of each production.

4 Model

The types of chunks and productions used in this model are not particularly novel. They are already used to model many phenomena and reflect a common approach in the ACT-R community. Arithmetic problems are represented as chunks with four slots: one for the operator, one for each operand, and one for the answer. An arithmetic problem will be placed in the goal, with the operator and operands specified and the answer slot empty. The **Retrieval** production retrieves from long-term memory a chunk matching the goal and copies the answer back to the goal. If retrieval is not possible, a backup strategy must be used. The **Iteration** production implements one such strategy by creating a subgoal to compute the answer iteratively, by counting for addition problems and adding for multiplication problems. As in the case of retrieval, the subgoal then returns the answer to the parent goal. Problems involving 0 as operand are solved by the special-purpose production **Zero**. When the answer has been determined, it is then output and the goal is popped by the **Answer** production. The goal, which now contains the answer, then becomes a new chunk in long-term memory, or reinforces an existing chunk if an identical one already exists. This repeated reinforcement will raise the activation of the arithmetic facts until they can be retrieved reliably.

Since past goals are the only source of chunks (other than for environmental encoding), this technique of solving a problem by pushing a goal which can be solved either by directly retrieving the answer from the corresponding fact or by using a number of backup strategies (including computing the answer but also looking it up in a book or asking the teacher) is a general ACT-R technique to model problem-solving. By gradually raising the activation of the necessary facts with practice, it provides a general account of the transition from general problem-solving strategies toward more efficient ones. As noted in Anderson and Lebiere (1998), ACT-R essentially implements Logan's (1988) proposal for transition from algorithmic solutions to direct retrieval.

Generally, children will choose to retrieve more often for smaller problems and choose to compute more often for larger problems (Siegler 1988). It would of course be possible to learn the expected gain of the production implementing each strategy, but that expected gain would not be sensitive to the problem size. While in general people may use a complex procedure to choose between strategies, in this model it turns out not to be necessary. Instead, retrieval is always attempted first and only if it fails is the computation strategy selected. Since smaller problems are more frequent and therefore more active than larger ones, they will be retrieved more often. Conversely, retrieval will fail more often for larger problems, and the answer will then have to be computed. This preference for retrieval as the first way of solving the goal can be seen as an instance of the Obligatory Retrieval Assumption of Logan (1988).

5 Subsymbolic searning

Clearly, the activation of chunks storing arithmetic facts is going to be very critical to ACT-R's performance in cognitive arithmetic. The activation of a chunk is given as the sum of a base-level activation and an associative activation according to the Activation Equation. The base-level activation will change with experience according to the Base-Level Learning Equation in such a way that it grows approximately as a log function of the amount of practice. The strengths of association will change with experience according to the Posterior Learning Equation such that it will vary approximately as a log function of the odds of the chunk i being needed when j is in the environment. These activation quantities are converted into match scores that reflect the effects of partial matching through the Match Equation. In the case of a perfect match, the match score is just the activation, but in the case of a mismatch a penalty will be subtracted from the match score. There is noise in these match scores because of activation noise. If the match score is above a threshold the chunk will be retrievable and the probability of it being retrieved is described by the Retrieval Probability Equation. If there are multiple possible chunks that might match, the one chosen is the one with the highest match score and the probability of any one being chosen is described by the Chunk Choice Equation. Finally, match scores determine latency through the Retrieval Time Equation.

Errors can be committed whether the subject is computing or retrieving. Let us consider the example of the problem $2 + 3$. Because of ACT-R's partial matching process it is possible for ACT-R to retrieve an arithmetic chunk (e.g., $2 + 4 = 6$) other than the correct one. It is possible that even after the mismatch score is subtracted off, the wrong chunk will have the highest match score and be retrieved and its answer stored in the current goal. Errors can also occur using the backup procedure when the iteration subgoal returns an erroneous answer because of the misretrieval of a fact used in the procedure. The erroneous answer will also be stored in the goal. In both cases of retrieval and computation errors, not only will the answer to this particular problem be wrong, but the goal holding the incorrect answer is popped and becomes an erroneous long-term fact (here, $2+3 = 6$). This fact can then be retrieved as the answer to future problems and perpetuate the error. This otherwise correct retrieval of an erroneous fact becomes another source of error. This is quite similar to the approach of Siegler (1988) which also involves competition between memories for both correct and erroneous answers. It might seem possible that ACT-R could reach an unfortunate state where it has so practiced the wrong facts that it comes to believe them. Indeed this can occur and the next section describes what must be true for ACT-R to avoid getting absorbed into such error states.

6 The dynamics of cognition

Cognitive arithmetic performance increases over the years from marginal (less than 50% correct retrieval of small addition facts among four-year-olds as reported by Siegler and Shrager (1984), and even much worse for larger facts) to almost perfect and efficient retrieval for most adults under normal circumstances. At some point, children largely stop using computation to answer their arithmetic problems and just retrieve the answer, even though they still commit a relatively high percentage of errors. What happens when a child starts retrieving answers subject to these errors and stops getting regular feedback on their arithmetic performance? Can these errors be reduced through sheer practice at retrieval? It seems at first that the answer is no. To see that, let us concentrate without loss of generality on the base-level activation and ignore spreading activation and mismatch penalties. If the competing chunks C_1 and C_2 have the same lifetime L and are rehearsed with respective frequencies p_1 and p_2, then from the Optimized Learning Equation the difference in activation between them will be:

$$A_1 - A_2 = \ln \frac{p_1}{p_2}$$

Thus, with more practice these chunks will become more active, but assuming that the relative presentation frequencies of the two chunks are unchanged, their difference in activation will remain constant. The Chunk Choice Equation then implies that their respective probabilities of retrieval are also unchanged, which would mean that the percentage of misretrieval errors would remain constant instead of gradually declining. This analysis however makes the fundamental mistake of viewing learning as a static process exclusively driven by the environment and ignores the dynamic nature of the retrieval process. Strong chunks will tend to be retrieved more frequently than weaker ones and thus will be reinforced more often, becoming even stronger. However, due to the stochastic nature of the retrieval process, weak chunks always have a chance of being retrieved and might gradually catch up to the stronger ones. To determine which process will dominate, one needs to formalize the dynamics of the retrieval process. In the case of two chunks C_1 and C_2, the Chunk Choice Equation can be rewritten to express the odds of retrieving C_1 as a function of the difference in activation between the two chunks:[3]

$$Odds_1 = \mathrm{e}^{(A_1-A_2)/s}$$

The previous two equations can be combined to give the odds of the current retrieval as a function of the past ratio of rehearsal of the two chunks (i.e., p_1/p_2):

$$Odds_1 = Ratio_1^{1/s} \qquad \textbf{Dynamic Odds Equation}$$

This equation shows that the current odds of retrieval are very sensitive to the activation noise level. If $s > 1$, the current odds of retrieval are closer to even odds than the past odds. This will ultimately lead to each chunk becoming equally likely to be retrieved, i.e., a fairly chaotic system. If $s = 1$, the current odds of retrieval are equal to the past odds. This does not imply that the retrieval odds will be fixed, but rather that they will drift randomly with experience, driven by chance and external events. If $s < 1$, then the odds of retrieval become more extreme, with one becoming arbitrarily large and the other becoming infinitesimal. This defines a winner-take-all dynamic: strong chunks (hopefully the correct ones) are more likely to be recalled, which will strengthen them further, while weak chunks (hopefully the wrong ones) will be increasingly less likely to be retrieved. Clearly this is the desired behavior, and this analysis imposes a theoretical upper bound on the value of the noise parameter in ACT-R if this behavior is to be achieved.

Assuming that the noise value is such that some chunks will gradually dominate the retrieval process, the next step is to determine the shape and rate of that process. Each new experience will be added by the declarative learning mechanisms to the statistics of past history. This incremental change in the history of retrieval odds can be expressed by a differential equation, which allows for two approximate solutions:

$$Ratio_1 \approx (cn)^{\pm 1} \qquad \textbf{Rehearsal Ratio Equation}$$

The past frequency ratio of retrieving either chunk gradually diverges according to a power law in the amount of practice with exponent -1 for the loser and +1 for the winner (c is a constant which depends upon initial conditions and n is the total amount of practice). Combining this with the Dynamic Odds Equation, the current or observed odds of retrieving either chunk, and therefore the odds of commission errors, are a function of the amount of practice to the power of the inverse of the noise measure:

$$Odds_1 \approx (cn)^{\pm 1/s} \qquad \textbf{Retrieval Odds Equation}$$

Another way to view the Retrieval Odds Equation is in terms of the number of training examples needed to reach a particular accuracy. The number n of presentations of a particular problem needed to lower the odds of confusion errors below a threshold ε grows as a power function of the accuracy threshold to the exponent of the noise level:

$$n = 1/c\varepsilon^s$$

Up to now, this section analyzed the competition between two chunks that shared the same context but differed in their base-level activation, e.g., the correct fact 3 + 4 = 7 and an incorrect fact 3 + 4 = 8. One can extend the analysis to the competition between two correct facts, 3 + 4 = 7 and 3 + 5 = 8, which are both rehearsed regularly because they are correct answers but differ in their context, i.e., 3 + 4 vs. 3 + 5. One can show that the determining factor in their retrieval becomes the strengths of association between the part of the context in which they differ, i.e., 4 vs. 5, and the chunks. Since those strengths of association are weighed in the Activation Equation by their source level W_j and assuming a total source level W of 1, these strengths are then multiplied by a factor of 1/3 since for retrieval purposes the context of an arithmetic problem is composed of three sources: the operator and the two operands.[4] The Dynamic Odds Equation then becomes:

$$Odds_1 = Ratio_1^{1/3s}$$

The actual noise level has been divided by the 1/3 attentional weight, which means that the noise will have to be less than 1/3 for the correct answer to emerge, and the rate of convergence in the Retrieval Odds Equation will likewise be divided by 1/3. Generally, this implies that the more complex the problem, i.e., the more sources of activation in the context, the lower the noise level needs to be to guarantee convergence and the slower convergence will be.

One can extend this analysis to account for a number of additional factors (see Lebiere 1998 for details). The mismatch penalty can be shown to leave the rate of convergence unchanged but multiplies the ratio in the Dynamics Odds Equation by a factor equal to the exponential of the activation penalty. One can also generalize the analysis to more than two chunks. It can be shown from the Chunk Choice Equation that the odds of retrieving one of many alternative chunks is the harmonic average of the pairwise odds of retrieving that chunk over any other. Therefore, the same dynamic applies in which the strongest alternative will get increasingly dominant over all others since it dominates each independently. This result is a variant of Luce's Choice Axiom (Luce 1959). Finally, one can study the impact of external interaction such as teacher instruction on the dynamics of the system. While teacher correction has a major impact

[3] The symbol s is used in the following analysis instead of the symbol t from the Chunk Choice Equation to emphasize its origins in the chunk activation noise.

[4] Divding source level equally among goal components is a basic ACT-R assumption. However, Anderson and Reder (in press) have recently questioned that assumption. While operator and operands might have different source levels, it leaves the gist of this analysis unchanged.

on the Dynamic Odds Equation early on in the process, it becomes overwhelmed by the weight of experience if one allows the system to run uncorrected for a long time. This may be why ingrained errors are so hard to root out from human cognition. Error correction will still be possible later on, but a much larger amount of correct feedback will then be necessary to reverse the odds in favor of the correct solution. This need to keep the system relatively stochastic early on in the learning to prevent the odds from growing large (and therefore less susceptible to correction) suggests a positive effect of activation noise upon long-term accuracy. By keeping the process sufficiently random in the early stages, it prevents an occasional error (random or otherwise) from being locked in as the dominant answer too quickly and allows more time for the correct answer to emerge. Noise therefore performs a function similar to simulated annealing in a Boltzmann machine. In other words, noise is not (only) a shortcoming of the system but an essential contribution to its robustness in an imperfect environment.

As a final comment, the power law form of the Rehearsal Ratio and Retrieval Odds Equations can also be found in the evolution of biological and technological systems between states of equilibrium (e.g., West and Salk 1987). This is a consequence of the fact that these systems follow power-law frequency distributions (e.g., Zipf's law, Pareto's law) similar to those of the cognitive environment (Anderson and Schooler 1991).

7 Associative interference and its implications

The preceding sections describe the subsymbolic learning taking place in the model and how it accounts for retrieval errors. Chunks stating the same problem but different answers (e.g., 3 + 4 = 7 vs. 3 + 4 = 8) will compete primarily in their frequency of retrieval. Since the maximum noise level needed for the right chunk to emerge is relatively large, the confusion between these facts will not constitute a major factor in retrieval errors. Chunks stating the correct answer to different problems (e.g., 3 + 4 = 7 vs. 3 + 5 = 8) will not primarily compete on rehearsal frequency because they are both correct facts that will be rehearsed almost as often despite a slight frequency inequality. What will distinguish them is the context in which they are most often retrieved, e.g., 3 + 5 = 8 will usually be retrieved with the sources 3, +, and 5 (and not 4) in the goal. Thus according to the Posterior Strength Equation those two chunks will have strong positive associations from the sources 3 and +, and either 4 or 5. In addition, due to the logarithm in the Empirical Ratio Equation, they will also have negative associations to the other source (5 and 4 respectively) since the frequency ratio is expected to be significantly less than 1. Moreover, from the form of the logarithm function, the positive strengths of association will be strongly limited in value while the negative associations will quickly grow unboundedly negative as the frequency ratio gets close to 0. Thus, because of their unlimited inhibitory potential the negative strengths of association from incorrect context elements will be more essential in achieving (near-) perfect retrieval than the positive associations or the base levels, both of which grow much more slowly.

This reliance on the strengths of association from unrelated context elements has a number of far-reaching consequences for this model and the architecture itself. The first concerns the basic nature of declarative memory retrievals. Complex computation subgoals such as those created by the **Iteration** production will require many retrievals of counting and addition facts. However, those subgoals contain many values, some of which are unrelated to the chunks being retrieved at any one time. But the presence of these unrelated numbers in the context will lead to the strengthening of the associations between them and the unrelated fact retrieved, which will prevent those associations from becoming strongly negative and thus will hinder further gains in performance. The solution is to subgoal retrievals instead of performing them directly. This corresponds to moving the retrievals from the left-hand side of productions to the right-hand side and pushing subgoals to perform them on the stack. This operation focuses on the retrieval to be performed by creating a new goal of the same type as the chunk to be retrieved and which only includes the activation sources necessary to the retrieval. Once the retrieval patterns have been subgoaled, a production must fire to perform the actual retrieval, complete the pattern, and pop the goal. These productions, which complete a goal by matching it to a chunk in memory, are very basic and can be found in many other models (e.g., Lebiere and Wallach, in preparation). Indeed, these productions are so basic and pervasive that it could be argued that they correspond to an architectural primitive similar to the Obligatory Retrieval Assumption of Logan (1988). In addition to allowing the strengths of association to achieve optimal predictiveness, this technique increases the modularity of problem-solving knowledge and through the subgoals creates a permanent declarative memory of each problem-solving step. Finally, it would be more neurally plausible. In ACT-RN (Lebiere and Anderson 1993), ACT-R's neural network implementation, a declarative retrieval is implemented by gating all the possible connections from central memory, where the goal is held, to declarative memory. Since the latter is quite large, this imposes a heavy representational burden. This subgoaling proposal would reduce declarative retrieval to a transformation local to the central goal memory followed by a straightforward broadcast access to declarative memory. This memory access is similar to the CAP2 connectionist control architecture (Schneider and Oliver 1991; Schneider and Pimm-Smith 1997).

Another problem with associative learning arises from the fact that unlike the matching process, spreading activation is independent of the slot in which a particular source of activation appears. One instance involves tie problems (e.g., 7 + 7), in which the same operand appears as a double source of activation. While because of that doubling those facts receive more activation from the operands than non-tie facts, they cannot develop negative connections from their operands to inhibit facts from the same table row or column, because they have only one distinct operand (in this case, 7 appearing twice) which appears in every fact of that row and column. Over time, the (constant) gain in spreading activation for tie facts is overwhelmed by the lack of inhibitory connections from their operand to neighboring facts, and tie problems exhibit an increasingly large percentage of errors. One solution to prevent this clearly undesirable consequence is to recode tie problems to explicitly indicate the redundancy and provide a differentiating source of activation. For example, the repeated operand in 7+7 will be replaced by a special chunk called **Double**, resulting in 7+Double. Negative associations can then develop between the source **Double** and non-tie facts, inhibiting errors on tie problems. This explicit re-encoding is consistent with ACT-R's theory of chunk creation (see Anderson and Lebiere 1998, pp. 23–24) and with the experimental findings of Eliaser, Siegler, Campbell and Lemaire (in preparation). They report that subjects spend more time encoding tie problems than regular problems and that non-tie problems exhibit better performance than tie problems in artificial problem sets where tie problems are the rule rather than the exception as they are in arithmetic. This suggests that the advantage enjoyed by tie problems in arithmetic in not intrinsic but instead a matter of explicit representation. Another instance of confusion in spreading activation involves so-called near-tie problems, e.g. 6 + 7, which show better performance than problems of similar size because their mirror problem (7 + 6) receives the same spreading activation and suffers little mismatch penalty, essentially giving them two chances to find the correct answer. A final example involves so-called corner problems in which one operand is much larger than the other, e.g., 1+8. The larger operand, i.e., 8, will prime the answer of another fact (1 + 7 = 8), which suffers little mismatch penalty (between 7 and 8) and thus leads to excessively high error rates. While these problems can be solved in ad hoc fashion, their omnipresence in this simulation indicates that the underlying assumption that sources spread activation independently of their position in the context is deeply questionable. This assumption is known in the field of machine learning as the Naïve Bayes Assumption (Mitchell 1997). It provides for a sometimes enormous computational simplification and has been quite successful in practical applications, and indeed in past ACT-R models. However, the limits of this assumption have been exposed by this simulation because of the fundamental dependency on the learning mechanisms.

How could this problem be resolved in ACT-R? While maintaining position-specific associations might be both computationally unfeasible and philosophically dubious, it might be possible to take advantage of the duality between the processes of activation spreading and partial matching and do away with strengths of association entirely. This would certainly be compatible with ACT-RN, where a slot value only affects the matching of that slot. One problem would be to account for the gradual improvement in retrieval performance resulting from increased practice. One possibility would be for the activation noise of a chunk to decrease with practice:

$$S_n = \frac{S}{1 + \log(n)} \qquad \textbf{Noise Reduction Equation}$$

where n is the amount of practice of that chunk and S is the initial noise level at chunk creation. It is straightforward to show that this would provide for a similar power law of practice as the Retrieval Odds Equation. Decreasing the noise of chunks over time with their amount of practice is closely related to the technique of simulated annealing in Boltzmann machines (Ackley, Hinton, and Sejnowski 1985; Hinton and Sejnowski 1986), since noise in ACT-R has an effect similar to temperature in Boltzmann machines through the same Boltzmann equation. However, there are differences as well. The Boltzmann distribution in ACT-R is merely descriptive, whereas it is also used in Boltzmann machines to control every local unit fluctuation. More fundamentally, simulated annealing in Boltzmann machines happens on a small time scale for every pattern presentation, whereas under this proposal it would be a long-term process, with the activation noise decreasing over a large time scale with each rehearsal. Finally, temperature in Boltzmann machines is a quantity global to the entire network, whereas every chunk in ACT-R would have a different noise as a function of its amount of practice, with well-settled knowledge being gradually frozen in place but more recent knowledge showing significant fluidity. Geman and Geman (1984) show for their Gibbs Sampler that the fastest annealing schedule assured to converge to the energy minimum (which corresponds to the maximum a posteriori (MAP) estimate of the underlying distribution) is of the form given by the Noise Reduction Equation.

8 Lifetime simulation

Lebiere (1998) presents very close fits of the model to the data presented earlier by assuming a certain distribution of knowledge strength at a particular point in time and a particular set of parameter values. While this method is widely used in Cognitive Science and often produces both tractable analyses and excellent simulation fits, it suffers from a number of disadvantages: it requires additional assumptions about the state of knowledge at particular points in time, it allows different parameter values to be estimated for each fit, and it provides only an incomplete understanding of the model's dynamic nature. In the lifetime simulation presented here, the same model is run with the same parameters to simulate each data set through the full development of arithmetic knowledge over time.[5] The challenge is whether the model can provide a good fit to the results given these additional constraints. The answer is affirmative.

The key assumption is that the frequency of problem presentation decreases with the size of its arguments. Based on the studies of textbook presentation frequencies of Hamman and Ashcraft (1986) and Ashcraft and Christy (1995), the model was exposed to 4000 problems per simulated year, with the largest problem being about 2.6 times less frequent than the smallest. Figure 5 presents the evolution of the problem-size effect over time at the simulation points for each grade:

The speeding up of response time for small facts mostly represents the effect of strengthening through practice. The speeding up for large facts represents the gradual switch from computation to retrieval as well as the increase in retrieval speed.

The addition retrieval data for four-year-olds is modeled by looking at the lifetime simulation after 1000 problem presentations, corresponding to about one fourth of a year of training (Fig. 6). The smaller percentage of correct retrievals for larger facts reflects the lower amount of practice

[5] The detailed model is available on the ACT-R web site at http://act.psy.cmu.edu

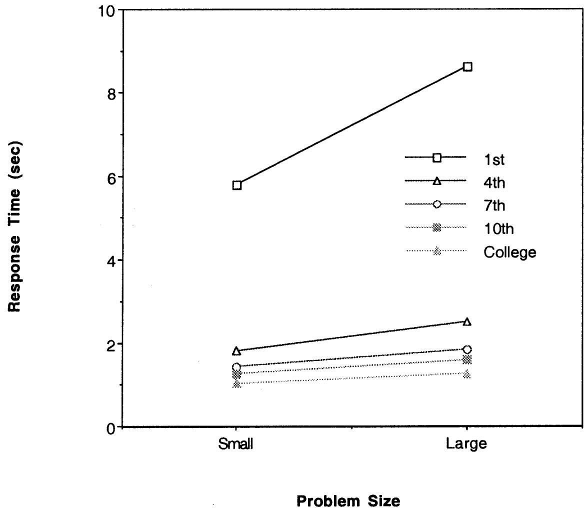

Fig. 5. Problem-size effect over time (simulation)

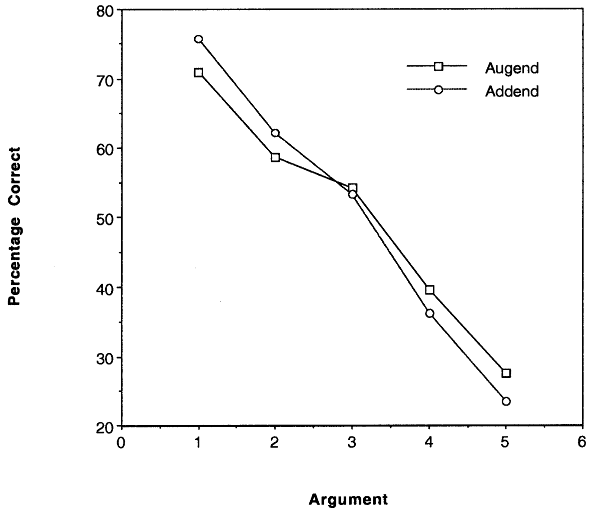

Fig. 6. Percentage correct in addition retrieval in the first cycle (1000 problems) (simulation)

received. Thus, they are both less likely to reach the threshold, and more likely to be invaded by a more active fact for a smaller problem. The asymmetry between addend and augend results from the use of backup computation during training: a large addend is more likely to result in a computation error because it involves more iterations. Those errors become facts when the goal is popped, which in turn can produce retrieval errors in later trials as is the case here.

The lifetime simulation also reproduces the problem size effect for both multiplicand and multiplier in the multiplication computation by repeated addition:

The multiplier effect is due to the number of iterations, while the multiplicand effect is due to the higher probability of errors for large facts. The very low percentage of errors for the multiplicand 5 results from the fact that repeatedly adding 5 only uses two single-digit addition facts ($0 + 5 = 5$ and $5 + 5 = 10$, which also happen to be a zero problem and a tie problem), which therefore were rehearsed more often in past computations than facts for other multiplicands which are more distributed. That extra practice translates into better performance in later trials.[6]

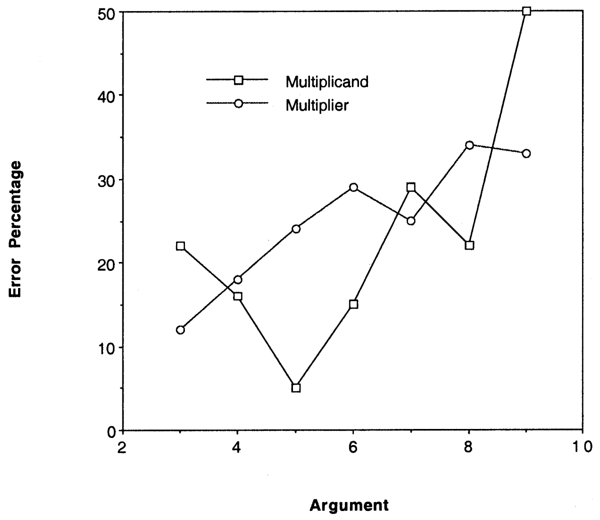

Fig. 7. Percentage errors in multiplication computation in cycle 3 (~4th grade) (simulation)

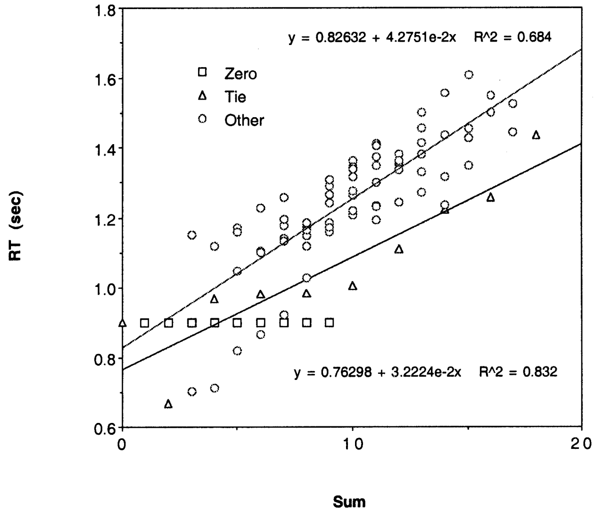

Fig. 8. Problem size effect at final state (simulation)

Figure 8 plots the response times at the end of the simulation (adulthood) for each problem category. The flat response time for zero problems is due to the use of the retrieval-free **Zero** production. The rest of the problems are overwhelmingly solved by the **Retrieval** production. The faster retrievals and lower slope for tie problems is due to the double-source effect. The problem-size effect generally stems as usual from the lower activation of larger facts resulting from a smaller amount of practice.

Finally, in Fig. 9 one can look at the behavior of the lifetime simulation to confirm the analysis predicting a power law decrease over time of the odds of retrieval failure (hence computation) and the odds of retrieval error.

The odds of computation and retrieval error indeed decay according to a power law, with exponents close to those

[6] An alternative explanation would be that 5 is a more prominent number, leading to additional practice of counting by 5. That assumption is unnecessary here.

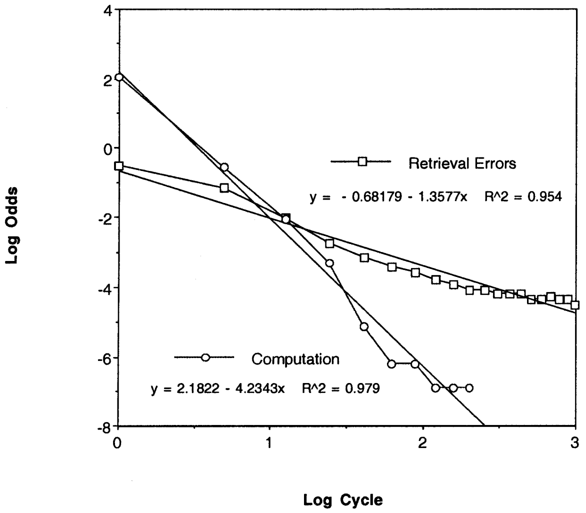

Fig. 9. Odds of retrieval error and of computation for addition problems as a function of practice (simulation)

predicted by the analysis. The curves for the multiplication facts display an even closer fit to the power law, with similar exponents. In addition to the data fits previously reported, these computation and retrieval error curves generally fit the data reported by Siegler and Robinson (1982), Siegler and Shrager (1984), and Siegler (1988) on the percentage of strategy use and retrieval errors at various points of development. Together, these results and those presented in the next sections provide strong corroboration of this model and ACT-R's theory of learning.

9 Choosing

This model always attempts to retrieve the answer and only computes when the retrieval fails. It is often assumed that human subjects decide which strategy to use based on the characteristics of the problem, in particular the percentage of success of each strategy. Siegler and Robinson (1982) and Siegler and Shrager (1984) report a very strong correlation between percentage of overt strategy use[7] on each problem and percentage of errors on those problems. The correlation is quite strong between percentage of overt strategy use and percentage of errors on retrieval trials, and still present but much weaker between percentage of overt strategy use and percentage of errors on overt strategy trials. They also report a strong correlation between percentage of overt strategy use on each problem and retrieval latency, and a weaker correlation between overt strategy use and latency of overt strategy.

For comparison, the lifetime simulation was run for about half a year of addition training, and then the correlation between strategy and errors was collected (Fig. 10).

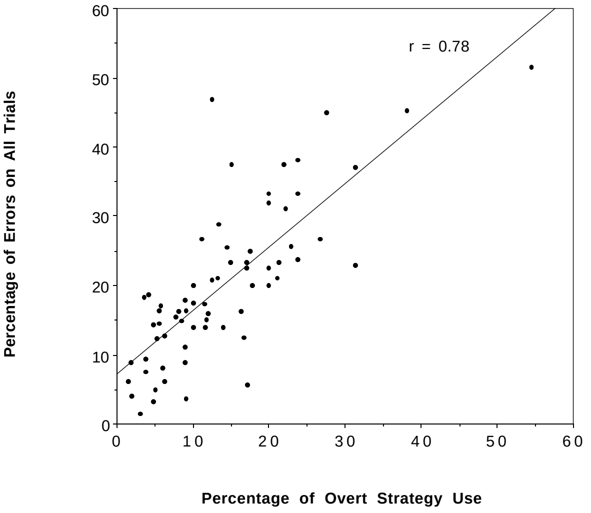

Fig. 10. Correlation between strategy use and errors on all trials (simulation)

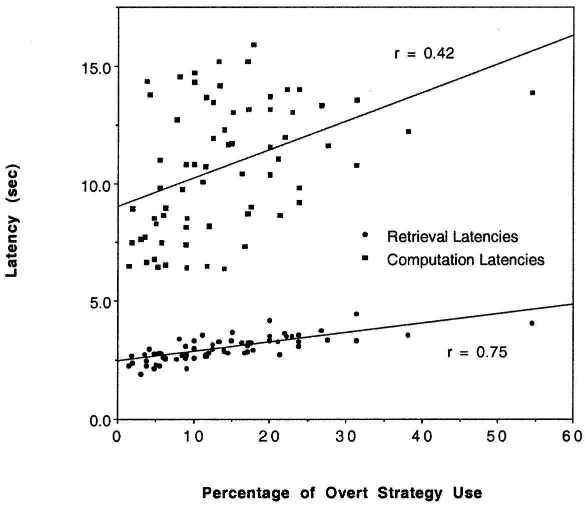

Fig. 11. Correlation between strategy use and latencies (simulation)

The correlation between percentage of overt strategy use and percentage of errors on all trials is quite high (0.78). The correlation between percentage of overt strategy use and percentage of errors on retrieval trials is slightly higher (0.81), whereas the correlation between percentage of overt strategy use and percentage of errors on computation trials is much lower (0.22). The correlation between the latency and the percentage of overt strategy use was also computed (Fig. 11). The correlation between percentage of overt strategy use and retrieval latency is high (0.75), and the correlation between percentage of overt strategy use and computation latency is somewhat lower (0.42). These correlations are quite close to the values in Siegler et al.'s subject data.

The lifetime simulation can reproduce those correlations, even though it does not perform any explicit choice of strategy because all measures tap into the same underlying variables, namely activation strength and problem complexity. Of course, this does not preclude a more elaborate model, in which the choice of a strategy would be made depending upon the characteristics of the problem and the strategy's past history of success, as Siegler and Shipley (1995) propose, but the choice between retrieval and computation can be made solely on the basis of activation.

[7] The term "overt strategy use" refers to trials where the subjects used an overt (i.e., audible or visible) computation strategy. For example, since the children were quite young they would put up fingers to count.

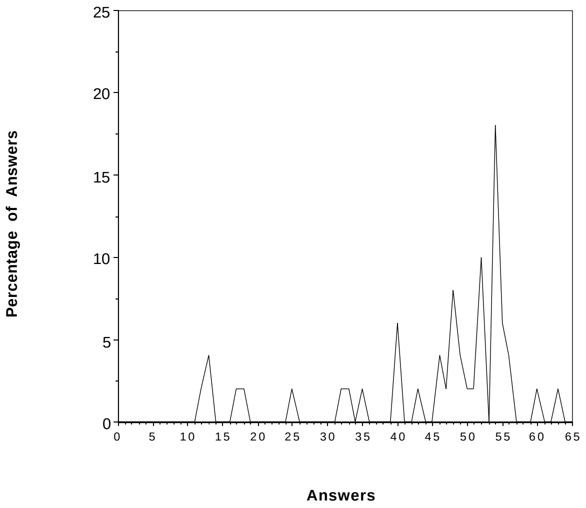

Fig. 12. Percentage of retrieval answers to 6 × 9 by 3rd and 4th graders from Siegler (1988)

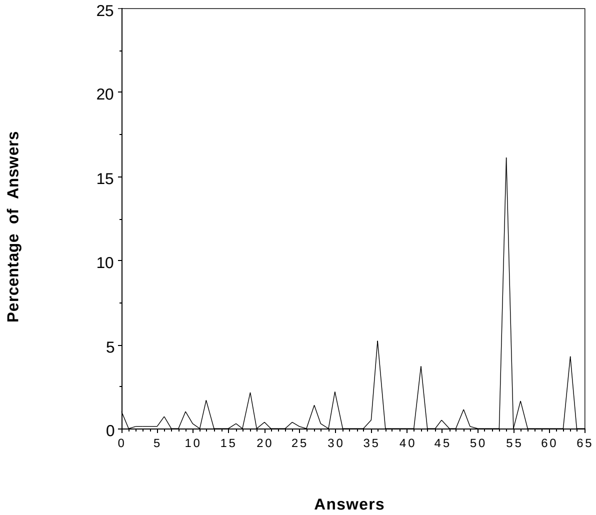

Fig. 13. Percentage of retrieval answers to 6 × 9 (simulation)

10 Guessing

The patterns of errors for multiplication retrieval are quite rich, but hard to examine systematically because they take place over a wider range of values and display some characteristics (like table errors and close misses) which are difficult to average and plot together. For those reasons, let us concentrate on the pattern of errors for a single problem. Siegler (1988) reports the answers to multiplication problems for an experiment in which third- and fourth-graders were instructed to state the answer to the problem without resorting to any explicit strategies. Figure 12 plots the percentage of answers to the problem 6 × 9. The correct answer, 54, is also the most likely one but only constitutes fewer than 20% of all answers. As in the case of addition errors, most of the errors are smaller than the correct answer, with the percentage of answers generally decreasing with the distance from the correct answer. Some of the errors can be classified as table errors, i.e., the answers appear in the same row or column of the multiplication table as the correct answer, e.g., 48 = 6 × 8. But most errors are not table errors, either because they are answers to facts elsewhere in the multiplication table (40 = 5 × 8 is neither on the same row or column as 6 × 9) or because they do not appear as answers in the single-digit multiplication table at all (46 or 52).

Figure 13 plots the percentage of retrieval answers to the same problem at an equivalent point in the lifetime simulation:

One positive result is that the percentage of correct answers corresponds closely to the data. Most of the errors are also smaller than the correct answer and the percentage of errors tends to decrease with the distance from the correct answer. This is to be expected since smaller facts are more active than larger ones, and the mismatch penalty increases with the distance from the correct fact. Unlike the data, almost all the errors are table errors (56 = 7 × 8 is one of the rare exceptions) which is a result of the dominant role played by the strengths of activation. There are also few close misses. About half the time no fact would reach the retrieval threshold and the model would fail to answer. Of course, subjects, too, have poorly established multiplication facts at that point in their development. But since they were instructed to state an answer, it is reasonable to assume that some of them were estimating or guessing the answer. This would explain the range of errors, the close misses; and the low percentage of table errors. How could ACT-R guess? Random guessing would not generate the pattern of errors reported, and repeated sampling or free association would still result in a majority of table errors.

But it seems that when humans engage in this sort of estimation, they do not do so randomly but rather compensate for the lack of a specific fact by relying on a wider base of related facts. How could that reliance on a set of facts rather than a single one be implemented in ACT-R? Such a mechanism has been proposed by Lebiere and Wallach (in preparation) to perform a similar task, interpolation, in a number of control problems. The solution is to produce the answer that minimizes the mismatches between that answer and the answers from each specific fact, weighted by its probability of retrieval. Formally, the answer is the value V that minimizes the following quantity:

$$V = Min \sum_i P_i(1 - Sim(V, V_i))^2 \qquad \textbf{Estimation Equation}$$

where P_i is the probability of retrieving chunk i, as determined by the Chunk Choice Equation, V_i is the value specified by chunk i, and $Sim(V, V_i)$ is the similarity between values V and V_i. Thus the term in parenthesis is the dissimilarity between the values, i.e., the amount of mismatch between them. If the dissimilarity between the values is interpreted as the error, then the Estimation Equation can be viewed as a standard least-squared error method. The well-known result that least-squared error solutions can be shown under certain assumptions to correspond to maximum likelihood hypothesis (e.g., Mitchell 1997) provides the connection between this equation and ACT-R's Bayesian framework.

By using the Estimation Equation to guess an answer when no fact reaches the retrieval threshold, the lifetime simulation generates the following pattern of combined retrieval and guessing answers for 6 × 9 (Fig. 14). The proportion of correct answers is about right, which results from the

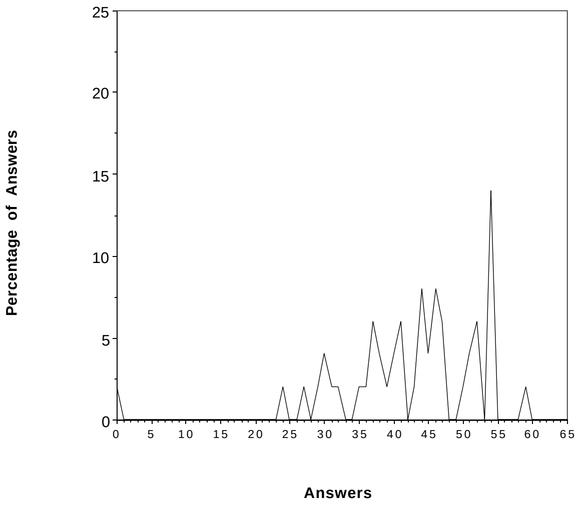

Fig. 14. Percentage of retrieval and guessing answers for 6 × 9 (simulation)

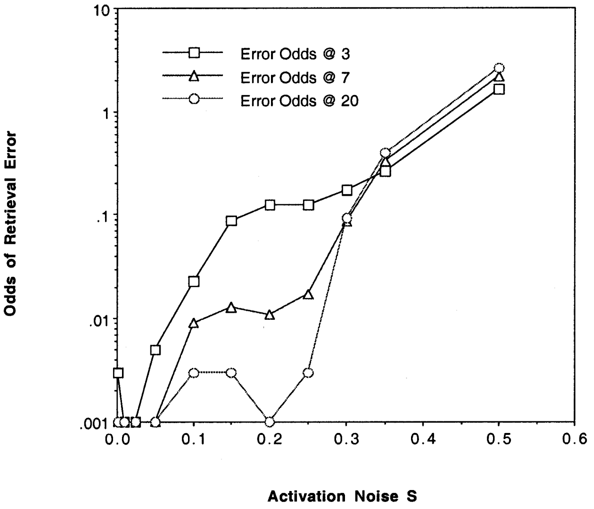

Fig. 15. Odds of retrieval error as a function of activation noise S

standard retrievals as shown in Fig. 13. Most of the errors are smaller than the correct answer, because smaller facts are more active, and thus have a higher probability of being retrieved and thus weigh more in the Estimation Equation. Most of the errors are not table errors, because estimation is a continuous process in which the answers to actual facts have little advantage over neighboring numbers. And there are many near-misses, with the probability of error decreasing with the distance to the correct answer. This results from the fact that while smaller facts are generally more active, facts closer to the correct answer will have a better match score because they incur a smaller mismatch penalty, and thus have a higher probability weight in the Estimation Equation.

The Estimation Equation is similar to an algorithm known in machine learning as the Bayes optimal classifier (Mitchell 1997). Instead of applying the Maximum Likelihood or Maximum A Posteriori hypothesis to categorize a new instance, the Bayes optimal classifier weighs all the hypotheses according to their posterior probabilities, and combines them to produce the most probable classification. As its name implies, this algorithm cannot be outperformed by any other classification algorithm that uses the same hypothesis space and prior knowledge. While it is generally considered too computationally expensive to be of direct practical use, its implementation in ACT-R is not prohibitively slow, and its behavior seems to correspond closely to the human ability to operate gracefully in continuous environments. And of course, as Mitchell (1997) points out, more practical algorithms can often be found that asymptotically approximate the characteristics of less feasible but optimal standards such as the Bayes optimal classifier.

11 Parameter sensitivity and optimality of cognition

The simulations described here are controlled by a number of real-valued parameters whose values were chosen to try to maximize the fit to the experimental data. While some amount of parameterization is needed in any architecture to account for individual and experimental variations, estimation of unknown variables (such as previous knowledge) and the like, there has been recently a concerted effort (Anderson and Lebiere 1998; Anderson, Bothell, Lebiere, and Matessa 1998) to understand the effect and constrain the values of ACT-R's global parameters. This section will examine the sensitivity of the model to its parameters, including global parameters such as the activation noise, retrieval threshold, and mismatch penalty, as well as domain-specific parameters such as training schedule, problem distribution, and feedback strategies. The formal analysis established that the activation noise is the main parameter controlling the speed of convergence to correct fact retrieval. Figure 15 plots the odds of retrieval error (on a log scale) over time (roughly 2nd grade, 6th grade and adulthood) for a range of activation noise values.

The results confirm the theoretical analysis with a convergence to almost perfect retrieval (less than 1% error) for noise values less or equal to the simulation value of 0.25, and gradual leveling or even increase for larger noise values. Large noise values, however, increase the probability that a weak fact would reach the activation threshold earlier, and thus lead to a faster transition to retrieval. Figure 16 plots the odds of retrieval as a function of activation noise.

This earlier transition to retrieval will lead to stronger reinforcement of the facts and therefore to faster retrievals as well. Thus the lifetime simulation noise level of 0.25 can be seen as optimal in providing the earliest and fastest possible arithmetic retrieval under the constraint of ensuring convergence to the correct answers.

A similar analysis applies to the retrieval threshold. Obviously, a higher threshold will delay the transition to retrieval as shown in Fig. 17.

But whereas a lower threshold will lead to earlier retrieval, it will also cause problems by allowing a rich-get-richer dynamic to take hold quickly: the chunks which are strong early invade the weaker ones which have not yet had a chance to build activation, leading to ingrained errors. Retrieval thresholds below the simulation value of −3.75 will lead to a leveling and even a gradual increase in errors over time. Larger threshold values will converge to the correct answers but more slowly because by delaying re-

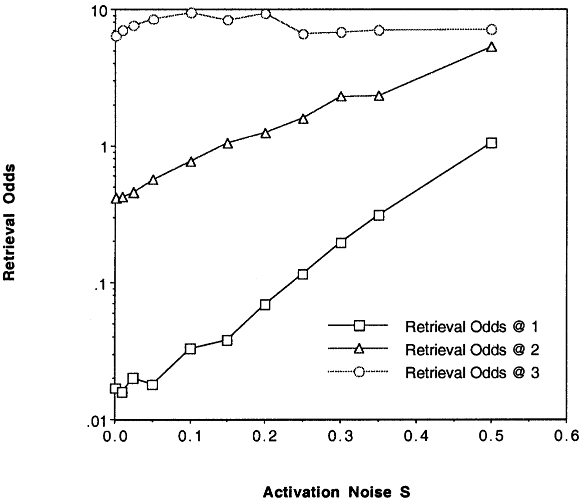

Fig. 16. Odds of retrieval as a function of S

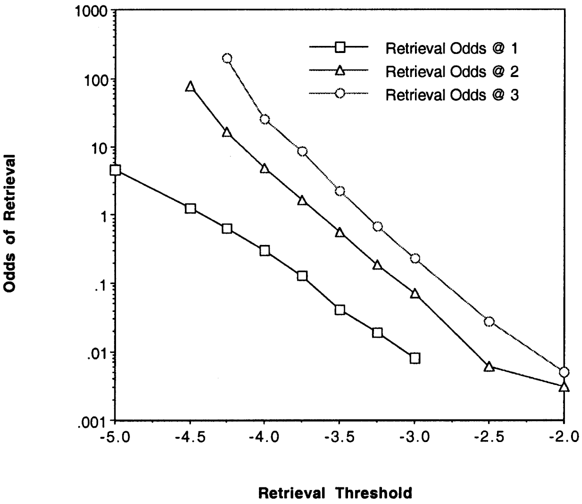

Fig. 17. Odds of Retrieval as a function of retrieval threshold (RT)

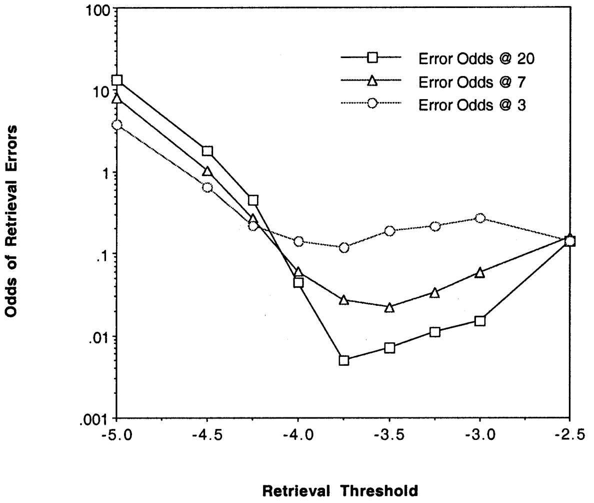

Fig. 18. Odds of retrieval error as a function of RT

trieval they rely more heavily on error-prone computation strategies which introduce their own errors. Thus the retrieval threshold used in the simulation is also optimal in an even stronger sense than the activation noise (Fig. 18).

A similar analysis can be performed for the mismatch penalty (see Lebiere 1998 for details). Mismatch penalty values smaller than the simulation value lead to error odds increasing over time, whereas larger values lead to slightly faster convergence to correct answers. Those larger values, however, lead to somewhat slower transitions to retrieval, making the mismatch penalty value optimal in the same sense as the noise. One can also analyze the impact of the domain-specific parameters. The density of the presentation schedule, i.e., the average delay between problems, has a similar impact as the retrieval threshold since its effect from the Optimized Learning Equation is simply to raise or lower the base-level activation by a constant amount. The steepness of the frequency distribution, or how much more common small problems are than large ones, has relatively little impact on the odds of retrieval and errors, suggesting that while it is central to the problem-size effect it is not a fundamental cause of arithmetic learning difficulties. Since students will confront a skewed distribution of problems in the external world, teachers might as well match that distribution and optimize the students' performance on those problems that they are most likely to encounter. Finally, as the theoretical analysis predicted, convergence to the correct answer is very sensitive to the initial feedback probability, emphasizing the need for constant feedback at the start of the learning curve.

The general conclusion is that while the qualitative behavior of the model is preserved across a range of parameter values, the values for each major parameter used in the lifetime simulation can be shown to produce optimal behavior. While this might not be too surprising for the domain-specific parameters controlling the teaching of arithmetic, which might have been selected for precisely the purpose of optimizing learning, it is extremely unlikely that the parameters of the human cognitive system were optimized to perform arithmetic. Human cognition developed in a constantly changing, uncertain world in which stochasticity and approximation were desirable qualities. That those values would be optimized for an abstract, exact, and unchanging task such as arithmetic is quite unlikely. One possibility is that our form of arithmetic (e.g., a base-10 system instead of a base-60 system) was developed to fit the capacities of the human cognitive system. Another, perhaps more likely, possibility is that human cognition might be even more adaptive that had been assumed, either in a parametric or strategic sense.

12 Conclusion

While other ACT-R models have been more complex (e.g., Schunn and Anderson 1998), the lifetime simulation of cognitive arithmetic stretched the limits of the architecture in a number of directions: its length (tens of thousands of problems taking hours of simulation), its reliance on learning, and its dynamical nature. While those dimensions combined to make this model considerably more difficult to develop than usual, the constraints provided by this approach yielded a number of contributions:

1. A precise account of a number of central results in the field of cognitive arithmetic. While the model is broadly consistent with previous activation-spreading theories of cognitive arithmetic, its basis in a general-purpose Bayesian learning architecture provides a systematic account of the causes and conditions of these effects.
2. A number of practical lessons for the teaching of arithmetic. Because the model makes detailed predictions that are affected by every aspect of the simulation, it can predict which conditions are critical to learning (feedback, spacing) and which are not (regular frequency differences).
3. A number of lessons for the architecture, including the view of retrieval as subgoaling to limit the sources of activation to those critical context elements, and ways to correct the deficiencies of the assumptions behind the associative learning mechanism. The nature of the task, requiring a fairly simple model but a very long, mostly self-correcting, learning simulation, was essential in deriving these lessons.
4. A connection to the machine learning field. It suggests that despite its past success, a popular algorithm such as the Naïve Bayes classifier has limitations in modeling the full scale of human cognition. It also suggests that despite being viewed as too computationally expensive, a more powerful algorithm such as the Bayes optimal classifier in fact closely approximates some of the aspects of human performance, hinting at the possibility of an efficient neural implementation of that algorithm.
5. A study of the sensitivity of the model to parameter values, both architectural and domain-specific, which shows that they are in fact optimal for some measure of performance. Since the human cognitive system was presumably not developed to perform precise tasks such as arithmetic, this raises further questions about the actual limits of its adaptiveness.
6. A view of cognition as a dynamic system (e.g., van Gelder 1998). Unlike fixed models or models that learn exclusively from an external environment, the behavior of this model and its changes over time are primarily determined by its own operations, which follow internal dynamics that depend upon the fundamental parameters of the architecture. Because of the richness and variety of those non-linear dynamics, those models are better able to explain the full diversity of human cognition.

References

Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science, 9,* 147–169.

Anderson, J. R. (1990). *The adaptive character of thought.* Hillsdale, NJ: Erlbaum.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Erlbaum.

Anderson, J. R., Bothell, D., Lebiere, C. & Matessa, M. (1998). An integrated theory of list memory. *Journal of Memory and Language, 38,* 341–380.

Anderson, J. R. & Lebiere, C. (1998). *The Atomic Components of Thought.* Mahwah, NJ: Erlbaum.

Anderson, J. R. & Reder, L. M. (in press). The fan effect: New results and new theories. *Journal of Experimental Psychology: General*.

Anderson, J. R. & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science, 2,* 396–408.

Ashcraft, M. H. (1987). Children's knowledge of simple arithmetic: A developmental model and simulation. In J. Bisanz, C. J. Brainerd & R. Kail (eds.), *Formal methods in developmental psychology: Progress in cognitive development research* (pp. 302–338). New York: Springer.

Ashcraft, M. H. (1992). Cognitive arithmetic: A review of data and theory. *Cognition, 44,* 75–106.

Ashcraft, M H. (1995). Cognitive psychology and simple arithmetic: A review and summary for new directions. *Mathematical Cognition, 1,* 3–34.

Ashcraft, M. H. & Christy, K. S. (1995). The frequency of arithmetic facts in elementary texts: Addition and multiplication in grades 1–6. *Journal for Research in Mathematics Education, Vol. 26, No. 5,* 396–421.

Benford, F. (1938). The law of anomalous numbers. *Proceedings of American Philosophy Society, 78,* 551–572.

Campbell, J. I. D. (1995). Mechanisms of simple addition and multiplication: A modified network-interference theory and simulation. *Mathematical Cognition, 1,* 121–164.

Geary, D. C. (1996). The problem-size effect in mental addition: Developmental and cross-national trends. *Mathematical Cognition, 2,* 63–93.

Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 6,* 721–741.

Hamann, M. S. & Ashcraft, M. H. (1986). Textbook presentations of the basic addition facts. *Cognition & Instruction, 3,* 173–192.

Hinton, G. E. & Sejnowsky, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart, J. L. McClelland, and the PDP Group, *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 282–317), Cambridge, MA: MIT Press.

Kirk, E. P. & Ashcraft, M. H. (1997). Verbal reports on simple arithmetic: A demanding task. Poster presented at the 38th Annual Meeting of the Psychonomic Society.

Lebiere, C. & Anderson, J. R. (1993). A connectionist implementation of the ACT-R production system. In: *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society,* (pp. 635–640). Hillsdale, NJ: Erlbaum.

Lebiere, C. (1998). *The dynamics of cognition: an ACT-R model of cognitive arithmetic.* Ph.D. Dissertation. Technical Report CMU-CS-98-186. Also available on the web at http://reports-archive.adm.cs.cmu.edu/

LeFevre, J-A., Sadesky, G. S. & Bisanz, J. (1996). Selection of procedures in mental addition: Reassessing the problem size effect in adults. In *Journal of Experimental Psychology: Learning, Memory and Cognition, Vol. 22, No. 1,* 216–230.

Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review, 95,* 492–527.

Luce, R. D. (1959). Individual choice behavior: a theoretical analysis. New York, NY: Wiley.

Moussouris, J. (1974). Gibbs and Markov random systems with constraints. *Journal of Statistical Physics, 10,* 11–33.

Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill.

Newcomb, S. (1888). Note on the frequency of use of the different digits in natural numbers. *American Journal of Mathematics, 4,* 39–40.

Peterson, S. & Simon, T. (in press). Computational evidence for the subitizing phenomenon as an emergent property of the human cognitive architecture. *Cognitive Science*.

Raimi, R. A. (1976). The first digit problem. *American Mathematics Monthly, 83,* 531–538.

Schneider, W. & Oliver, W. (1991). An instructable connectionist/control architecture: Using rule-based instructions to accomplish connectionist learning in a human time scale. In: K. Van Lehn (ed.), *Architecture for Intelligence* (pp. 113–145). Hillsdale, NJ: Erlbaum.

Schneider, W. & Pimm-Smith, M. (1997). Consciousness as a message aware control mechanism to modulate cognitive processing. In: J. Cohen & J. Schooler (eds.), *Scientific Approaches to Consciousness: 25th Carnegie Symposium on Cognition*.

Schunn, C. D. & Anderson, J. R. (1998). Scientific discovery. In: Anderson, J. R. & Lebiere, C. (eds.) *The Atomic Components of Thought*. Mahwah, NJ: Erlbaum.

Siegler, R. S. & Robinson, M. (1982). The development of numerical understandings. In H. Reese & L. P. Lipsitt (Eds.), *Advances in child development and behavior* (Vol. 16, pp. 241–312). New York: Academic Press.

Siegler, R. S. & Shrager, J. (1984). A model of strategy choice. In C. Sophian (Ed.), *Origins of cognitive skills* (pp. 229–293). Hillsdale, NJ: Erlbaum.

Siegler, R. S. (1988). Strategy choice procedures and the development of multiplication skill. *Journal of Experimental Psychology: General, 117*, 258–275.

Siegler, R. S. & Shipley, C. (1995). Variation, selection, and cognitive change. In: T. J. Simon & G. S. Halford (eds.), *Developing cognitive competence: new approaches to process modeling.* Hillsdale, NJ: Erlbaum.

Van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences, 21(5)*, 615–665

West, B. J. & Salk, J. (1987). Complexity, organization and uncertainty. *European Journal of Operational Research 30*, 117–128.