Text Integration and Mathematical Connections: A Computer Model of Arithmetic Word Problem Solving

MARK D. LEBLANC
Wheaton College, Norton, MA
SYLVIA WEBER-RUSSELL
University of New Hampshire

Understanding arithmetic word problems involves a complex interaction of text comprehension and mathematical processes. This article presents a computer simulation designed to capture the working memory demands required in "bottom-up" comprehension of arithmetic word problems. The simulation's sentence-level parser and text integration component reflect the importance of processing the problem from its original natural language presentation. Children's probability of solution was analyzed in exploratory regression analyses as a function of the simulation's sentence-level and text integration processes. Working memory variables measuring the combined effects of concepts to remember and text integration inferences account for a significant proportion of variance in children's solution probabilities across the first four grade levels (K-3). Consistent with previous results from others, which highlighted the significance of small changes in problem wording, the simulation offers a process-oriented perspective as to why natural language presentation constrains the comprehension of mathematical relationships.

INTRODUCTION

Researchers in cognitive science and mathematics education have recently emphasized that it is not only mathematical problem-solving abilities that are at the root of children's difficulties with mathematical word problems, but also linguistic difficulties (Cummins, 1991; Kintsch, 1988; Reusser, 1990). That is, many difficulties occur at the stage of comprehension of the natural language statement of the problem. Most of these researchers, however, stop short of defining the relationship between working memory and...
the process of "translating" natural language into correct mathematical relationships or into a solution.

It is clear that word problems vary among themselves in terms of solution difficulty, especially for young children (grades K–3), and attempts to explain the locus of problem difficulty have evolved from classifying problem types to an interdisciplinary focus on various cognitive dimensions of word problem solution. Traditionally, solution difficulty was explained as a function of children's logicomathematical competencies, that is, children's difficulties with a certain problem were explained in terms of deficient logicomathematical knowledge (Briars & Larkin, 1984; Riley & Greeno, 1988) or in terms of deficient numerical structures (Case, 1985; Okamoto, 1992). However, a growing body of empirical and theoretical work has shown that children's difficulties are strongly related to deficient language and text comprehension strategies. Linguistic factors (Cummins, 1991; Lewis & Mayer, 1987; Stern, 1993), presentational factors (Davis-Dorsey, Ross, & Morrison, 1991; De Corte, Verschaffel, & DeWin, 1985; Staub & Reusser, 1995), and the locus of low-achieving students' eye fixations (Hegarty, Mayer, & Green, 1992; Verschaffel, De Corte, & Pauwels, 1992) have only recently been recognized as a level of representation at which misunderstandings can occur.

Lacking in that work, however, is a model that links potential sources of linguistic misunderstanding with the working memory demands that result during the comprehension of "mathematical language," especially when the demands are placed on young readers. Recent results concerning the effects of small changes in problem wording (Cummins, 1991; Davis-Dorsey et al., 1991; De Corte et al., 1985), the role of integrated propositions in memory (O'Brien, 1987; Trabasso & Sperry, 1985; van den Broek, 1988), and the importance of working memory as a bottleneck in the comprehension process (Cooney & Swanson, 1990; Fayol, Abdi, & Gombert, 1987; Fletcher, 1986; Just & Carpenter, 1992; Sweller, 1988) highlight the importance of understanding potential difficulties in word problem solutions as a function of the demands on working memory. Such a view requires a detailed specification of the processes that are needed for a student to arrive at mathematical connections and how those processes utilize natural language to facilitate such connections.

In order to address working memory theories in conjunction with the strong evidence relating word problem difficulty with linguistic factors, we present a computer simulation of word problem solving that is: (a) sensitive to small changes in problem wording and (b) monitors the storage and processing demands that change as a result. Although there is strong evidence relating word problem difficulty with linguistic factors and memory demands are known to impact comprehension performance in the text comprehension literature, there has not yet been a systematic investigation of two
critical areas in the process of solving word problems: (a) the parsing effort and (b) the cognitive load derived from memory demands and inference-making requirements. In this article, we present a detailed simulation of children's complete solution process (i.e., beginning with a left-to-right reading of individual sentences) for all addition and subtraction problem types: Change, Combine, Compare, and Equalize.

As discussed more fully in the following sections, our simulation focuses on early stages of processing in order to provide an information processing perspective as to why changes in problem wording facilitate solution success. The information processing and learning assumptions in our research reflect the idea that different kinds of natural language used to convey a problem situation have different effects on the ability to conceptualize the problem in terms of correct mathematical relationships. Specific problem wording can: (a) highlight certain mathematical relationships (e.g., direct references to previously mentioned sets of objects facilitate the process of text integration), and/or (b) lead to action-oriented interpretations of events or quantities that are easier to retain in memory. Children's ability to follow-up on explicit set references (or infer such references) as well as infer action-oriented interpretations from static descriptions are crucial steps toward recognizing the conditions that make an arithmetic operation appropriate for a given situation.

We test the hypothesis that solution difficulty is a function of (a) the memory load due to storage demands, (b) the number of inferences that are required in order to establish text connections between conceptual sets or instantiate arithmetic actions, and/or (c) the amount of work it takes to parse the individual sentences in word problems. As our simulation solves each problem, it monitors its performance in these three areas and records measures of its performance. These measures are then compared to actual proportions of correct solutions produced by children in grades K–3 on these same problems. In particular, we present two analyses to test our predictions. First, we monitor the working memory demands that occur during a complete solution of each word problem. For each problem in a benchmark set of problems, we monitor the text integration and working memory requirements by measuring (a) the average number of conceptual units that appear in memory, (b) the average number of inferences made, and (c) the "total" working load, that is, the sum of the average number of concepts in memory and the average number of inferences made. In the second analysis, we monitor the amount of work it takes to parse the individual sentences in each problem as defined by the theory of conceptual expectation-based parsing. This analysis explores the possibility that problems that students find difficult to solve contain sentences that are difficult for our parser to handle. For each sentence we measure: (a) the average number of conceptual expectations that are posted, (b) the average number of concep-
tual expectations that become satisfied (fire), and (c) the average number of conceptualizations that occupy working memory as the sentence is parsed.

The article proceeds as follows. We first propose a benchmark of phenomena that any simulation of word problem solving must attempt to account for. Following examples that highlight our simulation's capabilities, we present a detailed comparison of our processing assumptions with other word problem simulations. We then present two exploratory analyses that correlate children's probability of solution on a set of problems with the monitored performance of the computer simulation on that same set of problems. The results are discussed in terms of how our simulation meets or fails to meet the benchmark of phenomena presented next.

**DATA TO BE ACCOUNTED FOR**

In conjunction with developmental theories of children's addition and subtraction competencies (Briars & Larkin, 1984; Fuson, 1994; Okamoto, 1996; Riley & Greeno, 1988), a number of task characteristics individually and in combination have been shown to affect the solution probability of addition and subtraction word problems. This section presents a benchmark of phenomena to account for in a computer simulation of word problem solving.

Much of the focus of word problem research in the previous decade has been concerned with identifying the semantic relations in various types of word problems and how those relations map onto schema of part-whole knowledge. With respect to problem difficulty, a classification according to *semantic structure* is a widely accepted categorization of arithmetic word problems, where *semantic structure*, in general, refers to the semantic relations between sets in the problem statement. Initially proposed by Heller and Greeno (1978) and generally consistent with other classification schemes (Carpenter & Moser, 1982; Mayer, 1981; Nesher & Katriel, 1977), addition and subtraction word problems are classified into one of four types of semantic structure: Change, Combine, Compare, or Equalize. Examples are shown in Table 1 and a complete set of problem wordings is provided in Appendix A in Table 12.

Change problems include significant action language that describe situations that occur over time, such as changes in the location or possession of objects. For example, Problem 1 in Table 1 describes a transfer of objects from one person to another. Combine problems involve static descriptions of the numerosity of two disjoint (sub)sets and the union of those two sets, for example, Problem 2 in Table 1. Unlike Change problems, Combine problems do not contain significant actions. Compare problems involve the static comparison of the numerosity of two disjoint subsets, for example, Problem 3 in Table 1. Like Combine problems, Compare problems do not contain significant actions. Equalize problems are a hybrid of Change
### TABLE 1
Problem Classification by Semantic Structure

<table>
<thead>
<tr>
<th>Semantic Structure</th>
<th>Sample Word Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change</td>
<td>(1) Jacob had 5 soda cans. Then Kathy gave him 3 soda cans. How many soda cans does Jacob have now?</td>
</tr>
<tr>
<td>Combine</td>
<td>(2) Jacob has 3 Pepsi cans. Kathy has 7 Coke cans. How many soda cans do they have altogether?</td>
</tr>
<tr>
<td>Compare</td>
<td>(3) Jacob has 5 soda cans. Kathy has 3 more soda cans than Jacob. How many soda cans does Kathy have?</td>
</tr>
<tr>
<td>Equalize</td>
<td>(4) Jacob has 4 cans. If he finds 3 more cans, he will have the same number of cans as Kathy. How many cans does Kathy have?</td>
</tr>
</tbody>
</table>

and Compare problems (Carpenter, Hiebert, & Moser, 1981), which involve two distinct sets, one of which is changed to be the same in number (or equal) to the other set; for example, Problem 4 in Table 1. Equalize problems contain significant actions and describe situations that occur over time.

The empirical evidence has convincingly shown that the semantic structure of verbal problems strongly influences the difficulty of problems and the manipulative strategies that young children use to solve the problems. Several investigators (Cummins, Kintsch, Reusser, & Weimer, 1988; De Corte & Verschaffel, 1987; De Corte, Verschaffel, & Pauwels, 1990; Riley, Greeno, & Heller, 1983) have found significant differences in the probability of solution for problems both within a specific semantic type (i.e., performance is affected by the location of the unknown) and between these traditional semantic structure types; for example, the Compare problems are consistently the most difficult problems for children of age 5 to 9. On the average, Change and Equalize problems are easier than Combine problems, which are easier than Compare problems, although this must be qualified in the sense that there are differences in relative difficulty within these types, mostly due to the position of the unknown quantity (Hiebert, 1982).

In addition to semantic structure, recent empirical results have emphasized the importance of linguistic, presentational, contextual, and processing characteristics that affect children’s ability to successfully comprehend and solve word problems. In Table 2, we summarize these characteristics in a benchmark of phenomena to account for in a simulation of word problem comprehension. In general, the data associated with this benchmark motivated a shift in focus from the developmental acquisition (or absence) of part–whole knowledge to a focus on the effect that language presentation has on a mapping from text comprehension to part–whole knowledge. This change in focus is due largely to Cummins et al.’s (1988; Cummins, 1991) linguistic development view.
For example, in order for a simulation to account for potential errors due to misunderstandings of quantification or reference, such as (mis)interpreting the word "some" to be a qualitative adjective like "red," the simulation must provide access to individual lexical entries, as first suggested by Dellarosa (1986). Access to each lexical item becomes increasingly more important if a simulation is to be sensitive to slight changes in problem wording. As shown by Cummins (1991), De Corte et al. (1985), Staub and Reusser (1985) and others, problem rewording accounts for significant changes in solution probability. In addition, process-oriented views of text comprehension present new challenges for a simulation that solves problems from the same starting point as a child, that is, "reading" sentences in a word-by-word, left-to-right fashion. Eye-movement data show that successful problem solvers spend a significantly greater amount of fixation time on the most difficult types of relational language and that students spend more of their problem-solving time integrating between sentences than on an initial reading (Hegarty et al., 1992; Verschaffel et al., 1992). Capacity theories of comprehension highlight the importance of working memory as a bottleneck.

<table>
<thead>
<tr>
<th>Phenomena</th>
<th>Empirical Results to Simulate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Semantic structure</strong></td>
<td>Change ≤ Equalize &lt; Combine &lt; Compare</td>
</tr>
<tr>
<td></td>
<td>(&lt; means &quot;easier than&quot; with the assumptions that the location of the unknown is fixed and there is some overlap between subtypes).</td>
</tr>
<tr>
<td><strong>Misunderstandings of quantification or reference</strong></td>
<td>Reproduce common errors due to children's misinterpretations, e.g., if &quot;some&quot; is treated as an adjective like &quot;red&quot;, or &quot;altogether&quot; is interpreted as each.</td>
</tr>
<tr>
<td><strong>Presentational structure</strong></td>
<td>Solution probability decreases when sentences are not in chronological order or the protagonist is not always in the actor role.</td>
</tr>
<tr>
<td><strong>Changes in problem wording</strong></td>
<td>The presence of significant action language and/or explicit set reference language (&quot;of them,&quot; &quot;the rest&quot;) significantly improves solution success.</td>
</tr>
<tr>
<td><strong>General context and affective factors</strong></td>
<td>Use of familiar situations, favorite objects, and the names of friends facilitates a mapping between text and mathematical relationships.</td>
</tr>
<tr>
<td><strong>Locus of eye fixations and problem-solving time</strong></td>
<td>Time spent during an initial reading does not vary with problem difficulty.</td>
</tr>
<tr>
<td><strong>Demands on working memory</strong></td>
<td>Storage of previous information and processing of new information compete for a limited amount of capacity (activation).</td>
</tr>
</tbody>
</table>
in the comprehension process (Cooney & Swanson, 1990; Fletcher, 1986), especially when one considers the view that both storage and computational demands compete for a limited amount of working memory, as expressed by Just and Carpenter (1992):

Within any task domain large performance differences among individuals will emerge, primarily when the task demands consume sufficient capacity to exhaust some subjects' resources. In the domain of language comprehension, capacity limitations are more evident when the linguistic construction is more complex or when there is an extrinsic load. (p. 143)

Clearly, no simulation of word problem solution can account for all the characteristics presented in Table 2. However, it has become clear to us that in order to seriously address the linguistic development view in conjunction with working memory theories, a simulation must be sensitive to small changes in problem wording and monitor the storage and processing demands that change as a result.

THE COMPUTER SIMULATION

The simulation is currently implemented as a “bottom-up,” action-oriented problem solver. Bottom-up processing is opposed to a more global, “top-down” strategy that potentially involves instantiating schema in memory for each particular type of problem (e.g., a Compare schema) and/or an associated plan to look for quantities that can fill slots in that schema (cf. Marshall, 1991). Action-oriented processing maps all problems onto manipulative interpretations, even if the problem is to be solved without the aid of physical manipulatives. More specifically, our bottom-up, action-oriented implementation simulates a young reader or, at least, a novice word problem solver. We define a “novice” as one who comprehends and solves a word problem in a left-to-right (word by word), top-to-bottom (sentence-by-sentence), one-pass mode. In other words, a novice reads a sentence and then attempts to integrate the new information before proceeding to the next sentence. This definition is currently implemented in its strictest sense in that neither individual sentences nor entire problems are reread.

The simulation is composed of two components: EDUCE and SELAH. EDUCE is an expectation-driven parser adapted from the work of Schank and Riesbeck (1981) and developed according to principles found to be distinct for word problem solving (Burns, 1993; LeBlanc & Weber-Russell, 1989). EDUCE parses a number of language features inherent in word problems, including: (a) quantities (e.g., 3, three); (b) noun compounds

1 The simulation program is written in Common Lisp. EDUCE is a rule-based parser whereas SELAH handles text integration and memory management in a functional fashion. The program does not have a connectionist component.
(e.g., soda cans); (c) pronominal reference (e.g., she); (d) time sequence (e.g., then, in the beginning); (e) set partitions (e.g., by ownership: "Dick and Jane have"); by type of object: "8 cats and dogs"); (f) ellipsis (e.g., "Ed has 3 cans. He found 2 more." That is, 2 more cans); and (g) reference to previous sets (e.g., 4 of them). EDUCE parses each sentence in a word-by-word, left-to-right fashion, building a conceptualization of the sentence as it proceeds. After each sentence is completely parsed, EDUCE passes its conceptualization of that sentence to SELAH.

SELAH is a text integration component that integrates EDUCE’s canonical representations of individual sentences with any preceding ones already in working memory. Text integration may be requested explicitly based on EDUCE’s representation of a sentence or the integration is inferred. Based on the integration of explicit conceptual actions in the text or direct or implicit references between sets, SELAH constructs a situational or action-oriented interpretation of the problem (LeBlanc, 1993b). Once SELAH completes the integration of a new sentence, control returns to EDUCE to parse the next sentence. This repetition of parsing then integrating continues until there are no more sentences in the problem. Once the problem is completely read, SELAH selects an appropriate counting strategy to solve the problem.

An Example

Some of EDUCE’s sentence-level reading capabilities and SELAH’s text integration processes are revealed in the following annotated script of the simulation’s output while solving the following word problem:

Jacob and Kathy have 8 soda cans. Jacob has 3 of them. The rest of them are Kathy’s. How many soda cans does Kathy have?

EDUCE parses the first sentence, "Jacob and Kathy have 8 soda cans," and produces the "state of possession" conceptualization shown here:

CONCEPT: STATE: possession

POSSESSOR: ((+human) (group (Jacob Kathy)))

OBJECT: ((physical (-animate)

(function: contain(object: soda))

(quantity: 8))

We defer a detailed discussion of how EDUCE parses sentences until the more interesting second sentence; however, two points are worth noting concerning the process that leads to the conceptualization of the first sentence. First, EDUCE has interpreted the noun compound "soda cans"
to mean "an object whose function is to contain soda." In the context of natural language understanding, nouns modified by nouns are considerably more complex to handle than nouns modified by adjectives, such as "red cans." The second point relates to our processing assumption that EDUCE is currently using a "correct" lexicon. In the final representation for this sentence, Jacob and Kathy are (correctly) recognized as joint owners of a set. In lexical terms, this implies that EDUCE is using a sufficient understanding of the word "and" in the context of joint owners.

Having completed the parse of the first sentence, the conceptualization is passed on to SELAH. Because EDUCE has not detected any pronominal or set reference in the first sentence, SELAH knows that there is no explicit request for text integration. No implicit integration is possible as no other conceptualized sets currently reside in long-term memory (LTM), so SELAH recognizes the conceptual possession of a quantified object as a set and stores the new set in LTM:

LTM-(1)
STATE: possession
   POSSESSOR: (Jacob Kathy)
   SET: 
      OBJECT: (physical (-animate) (function: contain(object: soda)))
      QUANTITY: 8

At this point, control returns to EDUCE to parse the second sentence of the problem: "Jacob has 3 of them." Each word in the sentence is read one at a time starting from left to right. Upon reading the first word, "Jacob," EDUCE recognizes this word as a name and performs a referential search to see if Jacob has been previously mentioned. Because Jacob has been previously mentioned (in the first sentence), the already existing concept for Jacob is accessed and its conceptual meaning is loaded into working memory:

CONCEPTS: (1) PERSON -- (physical (+human) (DEFINITERef: Jacob))
REQUESTS: nil

There are no expectations or requests generated by the word Jacob; that is, the reading of this word does not expect (or request) any particular concepts to follow. The word "has" is then read and the conceptual entry for the word has is loaded into working memory (2) along with its associated requests:

---

2 The convention of referring to long-term memory (LTM) and short-term memory (STM) as "locations" serves as a way of talking about connectionist concepts, for example, "level of activation," in traditional symbolic terms. In this simulation, LTM represents those concepts that are not currently involved in the parsing or integration processes; that is, they have presumably "decayed" to a state whereby reactivation is necessary in order to reference them.
CONCEPTS: (1) PERSON -- (physical (+human) (DEFINITEref: Jacob))
    (2) STATE: possession
        POSSESSOR: ?
        OBJECT: ?

REQUESTS: (i) Who possesses? [search for human earlier in sentence]
           (ii) What is possessed? [search for an object later on]

Before reading the next word, each of the requests associated with the
word has is tested to see if it might be satisfied. The first request (i) is of
course successful because a search finds the conceptual representation of
Jacob in working memory (i.e., Jacob’s conceptualization is + human) and
thus the (1) Jacob-concept is merged into the POSSESSOR-slot of the (2)
STATE-possession-concept. The second request (ii) is unsatisfied because
no conceptual-object currently resides in working memory. Having read
and completed the processing for the first two words, “Jacob has,”
EDUCE has one concept and one outstanding request:

CONCEPTS: (2) STATE: possession
            POSSESSOR: ((+human) (DEFINITExref: Jacob))
            OBJECT: ?

REQUESTS: (i) What is possessed? [search for an object later on]

The next three words, “3 of them,” form a noun group with an explicit
reference to a previous set. In short, the “3” causes EDUCE to (a) enter
noun group mode, (b) instantiate a conceptual quantity, and (c) generate a
request to find an object-concept in this noun group. In the context of a
quantified noun group, the word of is interpreted to mean that the concep-
tual referent to follow (currently unknown) is the whole that possesses as a
part the conceptual object in this noun group. In terms of requests, the
word of expects a definite reference of a conceptual object to follow and if
that object is found, the quantified object in the current noun group is
“part of” a previously known set. The word them leads to a set reference of
the previous set of Jacob and Kathy’s 8 soda cans. The noun group “3 of
them” is eventually merged into an “object-concept” with a quantity of
three that is PART-OF a previously known set. The outstanding request (i)
from the word has, which is expecting an object, is now satisfied. The can-
object is merged into the OBJECT slot of the (2) STATE-concept.

A final conceptualization for the entire second sentence resides in working
memory:
CONCEPTS: (2) STATE: possession
   POSSESSOR: ((+human) (DEFINITRef: Jacob))
   OBJECT: ((physical (-animate)
       (function: contain(object: soda))
       (quantity: 3)
       (PART-OF: "previous set containing this object")))

Of specific importance in this sentence is how EDUCE represents the explicit "part of" reference generated by the "of them" wording. This sensitivity to slight changes in problem wording highlights the importance of "starting from the beginning." As discussed more fully later, such explicit wordings and their associated representations facilitate the processes of text integration that lead to a coherent situational representation of the relationship between sets. As a counterexample, a more traditional wording of the second sentence in this type of problem is: "Jacob has 3 soda cans." In this case, EDUCE would encounter no explicit set reference and thus would not be able to represent any connection between the two sentences. The connection between sentences would have to be inferred later by SELAH.

As shown in the detailed parse of the second sentence, EDUCE generates a PART-OF slot from the "of them" phrase and thereby indicates an explicit connection between the objects in the first sentence and the objects in the current conceptualization (second sentence). SELAH performs the following steps in order to integrate this new conceptualization from EDUCE with the previous set in LTM-(1). First, SELAH notes that the source of the objects in the second sentence is a previous concept; that is, the objects from the first sentence. SELAH searches LTM for the set that includes the "soda-can" type object. Finding the previous set of eight specific "soda can" objects, SELAH knows that the source of Jacob’s three cans is a unique quantified set (as opposed to the universal set of all soda cans). Given that the source of the three cans is the set of eight cans, SELAH associates this qualitative relationship with the mathematical relationship that the set of three is a "member of" or "part of" the set of eight. SELAH makes a new set of three in LTM-(2) and links the two sets together; that is, the set in LTM-(2) is PART-OF the set in LTM-(1).

1 In this example, the reference from the word *them* ensures that the two sets being integrated involve the same type of object. In general, when SELAH attempts to determine if two sets are related, the objects in those sets need not exactly match. SELAH performs the following constraint checks: Do the objects match exactly (e.g., cans and cans) or are they "like-types" (e.g., infer that cats are pets)? If the objects are an exact match (e.g., cans and cans), is one object more qualified than the other (e.g., Pepsi cans and cans) or are they equally qualified (e.g., big black cats and big black cats)?
LTM-(2)
STATE: possession
  POSSESSOR: (same Jacob as above)
  SET:
    OBJECT: (physical (-animate) (function: contain(object: soda)))
    QUANTITY: 3
    PART-OF: the SET in LTM (1)

Having successfully integrated the two conceptualizations, SELAH then translates the relationship between these sets into an action-oriented interpretation. Because there is no significant action mentioned in the problem wording, SELAH must match the integration with a manipulative scenario that would represent the relationship between these sets. We consider the need to do this extra computing analogous to making an inference. In this example, SELAH infers that this static PART-OF connection can be associated with a procedural situation where one set is SEPARATED-FROM another set. More specifically, because LTM-(1) is the source of objects in LTM-(2), the set in LTM-(2) can be "constructed" in a manipulative sense by separating three cans from the previous set of eight in LTM-(1).

The transition from statically integrated text to manipulative actions such as separating-from is a critical feature of SELAH's novice problem-solving process. The hypothesis is that novice problem solvers who are working without the aid of manipulatives continue to think of relationships between sets in terms of the actions upon those sets. Theoretically, the model proposes that young children first construct procedural or action-oriented representations in order to arrive at an arithmetic operator (for similar arguments, see Fuson, 1994; Reusser, 1990; and Weber-Russell & LeBlanc, 1996). Whereas some problems include explicit actions (e.g., giving) that facilitate children's success (cf. Briars & Larkin, 1984), many word problems do not contain action language and require additional processing to infer a procedural interpretation of statically expressed relationships. In the current example, SELAH infers a separating-from interpretation from a static part-of relationship between two sets.

The third sentence of the problem, "The rest of them are Kathy's," involves a similar link between the third sentence and the first; that is, Kathy's are PART-OF the set of eight. In addition, the word rest implies that a separating-from action has already occurred; that is, the rest are left behind. Thus, the SEPARATED-FROM action that was inferred in the previous sentence is reinforced in this sentence and the objects remaining from that action are explicitly marked as belonging to Kathy in LTM-3. In short, SELAH's current representation is:

LTM-2 SEPARATED-FROM LTM-1 resulting in LTM-3.
TABLE 3
Summary of SELAH's Processing

<table>
<thead>
<tr>
<th>Sentence</th>
<th>Processing summary</th>
<th>Concepts in memory</th>
<th>Number of concepts</th>
<th>Number of inferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>make set</td>
<td>[J &amp; K's 8]</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>explicit part-of:</td>
<td>[SEP J's 3]</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>infer SEP-FR</td>
<td>[FR J &amp; K's 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>explicit part-of:</td>
<td>[SEP J's 3]</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>reinforce SEP-FR</td>
<td>[FR J &amp; K's 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>explicit result</td>
<td>[REsult K's ?]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In addition to making sets and performing text integration, SELAH monitors some of its cognitive processing such as the number of sets to remember and the number of inferences that must be made. Table 3 summarizes some of SELAH's processing that led to this current representation.

The question in the last sentence focuses SELAH's attention on the unknown amount of Kathy's set. At this point, SELAH's representation has reached a level of abstraction where only the quantities and arithmetic action are of interest. For example, SELAH is no longer concerned with Kathy and Jacob as participants or soda cans as the type of objects. Focusing on the quantities and arithmetic action, SELAH selects an appropriate counting strategy, in this case, Counting-Down-From (Carpenter, 1985) to arrive at an answer of 5. For example:

"Ok, start at 8: 7 that's 1, 6 that's 2, 5 that's 3; the answer is 5."

EDUCE's Processing Assumptions

The bottom-up design of our simulation was strongly influenced by the need to start the problem-solving process at the same point as children, that is, by reading individual sentences in a word-by-word, left-to-right fashion. Most computer simulations of word problem solution (Briars & Larkin, 1984; Cummins et al., 1988; Dellarosa, 1986; Kintsch, 1988; Kintsch & Greeno, 1985; Marshall, 1991; Okamoto, 1996; Riley & Greeno, 1988) start directly with coded representations of the entire problem or artificial propositions of individual sentences. The exception is Reusser's Situational Problem Solver (1990), although that simulation deals solely with Change problems. In particular, EDUCE requires each lexical entry to include a detailed specification of its potential syntactic and semantic contribution, independent of any particular sentence or phrase. This enables EDUCE to simulate how
ambiguous propositions are constructed during the process of reading. Although ARITHPRO (Dellarosa, 1986) and subsequent simulation work (e.g., Cummins et al., 1988) has successfully shown that ill-formed propositions of critical mathematical words and phrases do not trigger rules that instantiate superschema representations (e.g., when the proposition for “more-than” is ill-formed on input, the comparison proposition frame is not instantiated), these simulations do not provide a level of granularity that allows an investigation of the processes that lead to the construction of mathematical meaning inherent in relevant propositions (e.g., “have-altogether” or “have-more-than”). For example, it was not until we attempted to parse relational phrases such as “have more/less than” that we began to appreciate the unique processing that is required in order to correctly interpret comparative language. It is not entirely clear how to best quantify the amount of cognitive load required by these sentence-level reading processes, but EDUCE has exposed the presence of processes that occur in parsing comparative sentences that do not occur for other types of problems, for example, (a) the influence of the qualitative interpretation of “more,” (b) unsuccessful referential searches, and (c) ellipsis. As an example, we discuss the first of these points here and return to the other two after presenting our results.

EDUCE parses sentences in a word-by-word fashion, thus, each word has an independent lexical entry. For example, the lexical entry for “more” is defined to mean that the quantity in the current sentence is “in addition to” some other quantity. In the CHIPS (Briars & Larkin, 1984) and ARITHPRO (Dellarosa, 1986) simulations, the lexical entry for “more” is merged into the construct “have-more-than,” thereby implying a sophisticated understanding of the word more. On the other hand, EDUCE’s interpretation extends the qualitative sense of the word that young children learn quite early, for example, “I want more cake” (Walkerdine, 1990), with an action-oriented sense of the word where quantities are “joined.” For example, given the two sentences “Pat has 3 shells” and “Bob gave Pat 2 more shells,” EDUCE interprets the second sentence to mean that “Bob gave Pat 2 shells and those two are in addition to [an amount not mentioned in this sentence].” It is important to note that the definition of the word more always implies a referent (i.e., in addition to [what]), however the increase does not necessarily involve the joining of or difference between quantities. EDUCE handles “more cake” and “2 more” by using the same lexical entry for “more,” but notes that the differing contexts imply different procedures. “More cake” implies an increase but not a quantitative join whereas “2 more” implies an increase and a quantitative join, although the referent of the increase may be unknown. Likewise, when encountering “more” in

4 Because the sentence, “Bob gave Pat 2 more shells,” does not explicitly describe a previous amount, EDUCE’s representation does not include a specific referent. During the process of text integration, SELAH notices the implied reference to [some previous amount] and infers that Pat’s original set of [3 shells] is the missing referent.
a relational expression (e.g., "more than . . ."), EDUCE continues to interpret this as an increase and quantitative join, although in relational phrases the referent is known (e.g., "more than she had"). More specifically, EDUCE understands the relational "more than" expression without resorting to a "large set, small set, difference set" schema (cf. Riley & Greeno, 1988). With EDUCE, the difference relation is mapped onto a procedural interpretation where the difference is interpreted in the sense of an operator rather than the cardinality of some set. In this sense, EDUCE simulates a level of linguistic competence closer to a novice's qualitative understanding of "more" rather than the expert schema-based understanding implemented in other simulations.

In addition to providing a fine-grained tool for investigating the contribution of individual words in ambiguous phrases, EDUCE is sensitive to small changes in problem wording, which have been shown to affect solution success (Cummins, 1991; De Corte et al., 1985; Staub & Reusser, 1995). Unlike previous simulations, this sensitivity allows our simulation to handle events that are not in strict chronological order as well as highlight the important role of phrases such as "of them" and "the rest," which make explicit references to previous sets. By starting with the words, EDUCE simulates how natural language can facilitate the construction of mathematical connections in memory.

**SELAH's Processing Assumptions**

In regards to the text integration component, SELAH maps all problems onto an action-oriented interpretation, regardless of whether the text contains significant action language. This is in keeping with our focus on simulating the novice problem solver, as we assume that most young children are not yet capable of instantiating and maintaining in memory the problem-specific superschemas and associated goals for filling slots in those schemas. SELAH's process of text integration (whether explicitly requested by EDUCE's representation or inferred) maps relations between sets onto one of the arithmetic situations of JOIN or SEPARATE-FROM. In problems containing significant action language, SELAH need not infer the situation; for example, a transfer of possession to an existing set is mapped onto a JOIN. On the other hand, SELAH must infer the action in problems without significant action language. For example, if one "has more of something," that implies a JOIN took place prior to the possession of "more," or if one "has a part of something" that implies a SEPARATION-FROM a set took place in order to possess the part. The centrality of our action-oriented perspective is largely based on the "action theoretic" approach of

---

Our focus on simulating novice problem solving contrasts sharply with Marshall's (1991) hybrid (connectionist and production system) simulation where a schema is instantiated on the first pass and then relevant pieces of the problem fill that schema on subsequent passes. Marshall's connectionist component (the first pass) successfully simulates college student performance on the task of classifying situations expressed in story problems.
Reusser's *Situational Problem Solver* (SPS, 1990). SPS possesses an elaborate text-processing component to distinguish the representation of the situation from the representation of the text, as advocated by Kintsch (1986). Reusser's is the first attempt to simulate the progressive and incremental process of transformation from text to situation to equation. In particular, SPS stresses that from a problem-solving (and instructional!) point of view, arriving at a representation of the situation in a problem is not a superfluous process, but rather a necessary one. Although SPS reads and solves a wide range of very complex problems involving active descriptions of changes in quantities, it does not address static and/or relational language problems (e.g., Combine and Compare types). Likewise, while we agree with Briars and Larkins (1984) that "a strong implicit cue for the action of joining is provided by the words together, altogether, in all, or by a superset word (e.g., children) following subset words (e.g., boys and girls)" (p. 263), their CHIPS simulation does not handle the most complex types of comparative language. Thus, SELAH examines the role of action-oriented representations in statically worded problems (i.e., problems that do not contain significant action language), as suggested by Stern and Lehrndorfer (1992).

In addition to SELAH's mapping from text integration to action-oriented representations, the simulation monitors the total storage demands and number of inferences that are required in order to solve each problem. This follows from our assumption that problem difficulty is a function of linguistic competence and working memory load. For example, in contrast to the CHIPS simulation, SELAH does not simulate actions by counting out and moving manipulatives, such as blocks (or "chips") on the table. In CHIPS, word problems are solved by acting them out with representations of physical counters. For example, the sentence "David has 5 baseball cards" causes CHIPS to put out five counters, each labeled as a baseball card belonging to David. If this sentence were followed by: "Then Pat gave 2 baseball cards to David," CHIPS would move two new counters into David's pile, update these counters as belonging to David, and make one pile of seven counters. Our simulation, on the other hand, assumes an understanding of cardinality and maintains each set in memory, even if they are joined. Thus in our simulation, David's five baseball cards are represented by the quantity five, rather than five individual items, and two unique sets are maintained in memory: the two transferred to David as well as his original set of five. This decision is in keeping with our focus on simulating the total working memory requirements that novices may experience as they solve problems in a bottom-up fashion, including the cognitive demands of remembering the label, role, and quantity of previous sets rather than off-loading those requirements by depending on physical aids.

A related processing assumption is that EDUCE and SELAH possess "expertlike" reading and computational skills and work with an unlimited working memory capacity. Put differently, EDUCE and SELAH as described
in this article do not represent a developmental model in the sense of Briars and Larkin (1984), Riley and Greeno (1988), and Okamoto (1996). This assumption reflects a particular theoretical focus rather than a limitation of the simulation. Our intent at this point is to simulate and monitor the working memory demands for a wide range of wording types given adequate linguistic and mathematical knowledge, as suggested by Kintsch and Greeno (1985). With respect to the parser, this means that the syntactic and conceptual knowledge for each word in EDUCE’s lexicon is “correct.” With respect to text integration, this means that SELAH may handle a relatively large demand on working memory on some problems (e.g., text integration inferences are required while previous sets are remembered) although it may experience relatively low demands on others. The present simulation focuses on the overall demands of bottom-up reading rather than a prediction of errors potentially associated with an overload of working memory and/or an insufficient level of linguistic competence.

By starting our investigations with sufficient knowledge rather than by simulating various levels of development, we do not mean to imply that we view simulating sufficient knowledge as the most important or central level of understanding. We do, however, view a sentence-level parser that is capable of “expert reading” as a critical component in establishing a baseline of problem difficulty, especially when problem difficulty is viewed as a function of linguistic competence and working memory load. To date, a number of simulations have proposed and empirically verified developmental levels of mathematical and numerical competence, but these have been carried out without reference to the reading process and its demands on working memory. For example, CHIPS (Briars & Larkin, 1984) simulates the following range of developing abilities: (a) associate a chip with a single role (e.g., it belongs to Pat); (b) associate a chip with a double role (e.g., it once belonged to Pat but now it belongs to David); and (c) rerepresent problems, such as recognizing that actions can be reversed in time or that two subsets can be interchanged to produce identical situations. As shown by empirical comparisons of CHIPS level of mathematical knowledge and children’s solutions, developmental advances in ability have a statistically significant effect on solution probability that is comparable to a change in grade level; for example, the ability to use double-role counters is much more complex than the use of single-role counters. A more interesting result from our perspective is their result that “the range of complexity for knowledge of special language cues is wider than the range of mathematical knowledge for these problems” (Briars & Larkin, 1984, p. 283). That is, knowing how to handle special language cues has more of an effect on solution success than a developmental advance in mathematical knowledge. By monitoring memory load in conjunction with sufficient levels of linguistic expertise, SELAH allows us to ask whether some problems are difficult because of a relatively high demand on working memory. We view this as a
critical extension to the linguistic development view. That is, even if children understand part-whole relations (Riley & Greeno, 1988) and even if they can map ambiguous words onto part-whole relations (Cummins, 1991), solution difficulty can be a function of the aggregate demands required by the presence or absence of specific problem wordings.

Lastly, the implementation of these problem-solving steps in a computer model reflects a particular text comprehension bias that deserves mention. Reading word problems requires a set of domain-specific text comprehension strategies that are not utilized in the process of reading other types of texts, as pointed out by Kintsch and Greeno (1985). For example, in natural language, numbers function as attributes of objects whereas in word problems, quantified noun groups must be abstracted to the point where the numbers are the objects of interest (Nesher & Katriel, 1986). Researchers refer to this ability to translate between and/or abstract from natural language and the formal language of mathematics in a number of ways: "playing the word game" (De Corte & Verschaffel, 1985), understanding "textual presuppositions" (Kintsch & Greeno, 1985), and possessing the "mathematics register" (Spanos, Rhodes, Dale, & Crandall, 1988). Having said this, we are not assuming that language and mathematics are separate components, with the linguistic component handled before the math component. Looking at the linguistic stage of word problem understanding is not just a matter of tackling on linguistic comprehension prior to mathematical considerations. Rather the math language is embedded in the natural language; for example, expressions such as "part of" have math meaning but are understood in more general natural language terms by readers outside of a math context. Such sentences are parsed as natural language statements would be in a non-word-problem context, but then have the more rigorous math meaning imposed on actions and relations. A further exception to generality (of the text integration component) is that inferences that might apply to nonmath contexts are not made. For instance, "John has 1 million dollars" would not trigger the inference John is rich, as it might in a nonmath context. This difference reflects the assumption that a reader with even a little word problem experience does not make such inferences, but that novices or students from outside the "word problem culture" may. Therefore, although the text integration process is general (e.g. in its resolution of anaphoric references), it becomes specific to math word problem solving in executing math inferences (such as separation), just as inferencing procedures in non-math applications differ according to the context (as realized, e.g., by particular differing scripts).

**MONITORING PERFORMANCE**

The next sections present the results of two analyses that correlate children's probability of solution on a benchmark set of problems with the monitored
performance of the computer simulation's two components, EDUCE and SELAH, on the same set of problems. Although the two components are both involved during any simulation, we present two separate analyses in order to focus on the monitoring capabilities specific to each component. Simulation 1 isolates global and local measures of text integration and working memory in the SELAH component that account for a significant proportion of the variance in children's solution probability. Simulation 2 explores whether EDUCE's sentence-level reading processes might account for some proportion of the variance in children's solution probability. In both cases, monitored performance represents the simulation's measure of its "workload" while a problem is being completely solved (parsed, integrated, quantified). Two separate analyses are presented in order to focus on the effects of each component individually. Simulation 1 presents SELAH's working memory measures while solving the entire benchmark set of problems and Simulation 2 presents EDUCE memory management and parsing effort while reading all sentences of those same problems.

**Simulation 1—Text Integration and Working Memory**

**Method**
Children's probability of solution was analyzed in exploratory regression analyses as a function of six predictor variables, where the predictor variables were measures of text comprehension processes as measured by the text integration component, SELAH.

**Materials**
The word problems solved by the computer simulation are a classic benchmark set of 18 problems from the Riley et al. (1983; Riley & Greeno, 1988) experiments. The problems are elementary addition and subtraction word problems containing sentences that describe two known quantities and a third unknown quantity. The problems are one-step problems; that is, they require only one arithmetic operation to arrive at the unknown quantity, where the required operation is either addition (+) or subtraction (−).

The solution probabilities of the 18 problem types are the results of previous research with children (Riley & Greeno, 1988; Riley et al., 1983). The participants in these studies were children from kindergarten, first, second, and third grades. The analyses presented in this article match the performance of our simulation with the Riley and Greeno (1988) data; that is, when problems were read to the children and they did not have manipulatives (e.g., blocks) available.

---

6 From an implementation perspective, there is no need to run two separate simulations. Both EDUCE and SELAH can monitor and output their results in the same run. As noted earlier, the processes that are monitored in each component are sufficiently independent to warrant separate presentations.
TABLE 4
Example of Scoring for AVGMEM and AVGINF Variables

<table>
<thead>
<tr>
<th></th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing summary</td>
<td>make set</td>
<td>INFER &quot;3 to K&quot;</td>
<td>explicit result</td>
<td>of SEP-FROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>are part-of 8;</td>
<td>explicit SEP-FR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concepts in memory</td>
<td>[Jacob's 8]</td>
<td>[SEP K's 3 FR]</td>
<td>[SEP K's 3 FR]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Jacob's 8]</td>
<td>[Jacob's 8]</td>
<td>[REsult Jacob's ?]</td>
<td></td>
</tr>
<tr>
<td>Number of concepts</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2.00</td>
<td></td>
</tr>
<tr>
<td>Number of inferences</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0.33</td>
<td></td>
</tr>
</tbody>
</table>

Procedure
The 18 problems were input in English to the computer simulation one at a time. For each problem, the simulation (EDUCE and SELAH together) read and completely solved the problem. In this analysis, SELAH printed out a record of its text integration and working memory processes. The output for each problem was then analyzed to obtain scores for the variables described in the next section.

Predictor Variables
The probability of solutions obtained in the Riley and Greeno studies were used in regression analyses as a function of six predictor variables. The variables represent computational measures of SELAH's working memory demands (storage and processing) as it solved each of the problems (e.g., storing conceptualizations of sentences in working memory and performing text integration across sentences). To indicate how scores of two of the predictors were obtained, we refer to the example presented in Table 4. Summaries of text integration scores for each variable for all 18 problems are given in Appendix A in Table 13. The computer simulation's predictors are discussed as either local (sentence-level) or global (average across all sentences) measures. A summary of the three global and three local variables is given in the context of solving the following Change problem:

Jacob had 8 soda cans. Then he gave 3 soda cans to Kathy. How many soda cans does he have now?
GLOBAL Variables

AVGMEM
This is the average number of conceptual units in memory. This is the running sum of the number of conceptual units that appear in the simulation's working memory at the end of each sentence divided by the number of sentences in the problem. In these simulations, working memory was unbounded, as opposed to setting it at some predetermined size (cf. Fletcher, 1986; Just & Carpenter, 1992). Of interest here is the storage demand during bottom-up reading. A conceptual unit is defined as (a) a single conceptual set (e.g., [Jacob's 8]); (b) an arithmetic action (e.g., JOIN or SEPARATE_FROM); or (c) an action and conceptual set(s) that have been "chunked" together. In Table 4, the number of concepts in working memory at the end of the first sentence is one: [Jacob's 8]. In the second sentence, the presence of the arithmetic action SEPARATE_FROM in conjunction with two conceptual sets with known quantities results in a chunking (or merge) of three concepts into two concepts. Specifically, the three concepts, "[Kathy's 3] [SEPARATED_FROM] [Jacob's 8]," are chunked into two concepts: "[SEPARATE Kathy's 3 FROM] [Jacob's 8]." Two conceptual sets and an associated action are chunked only if both sets have known quantities; that is, [Kathy's some] [SEPARATED_FROM] [Jacob's 8] remains three concepts. Performing such chunking when the quantities are unknown reflects the fact that all the information needed to solve the problem (i.e., quantity-action-quantity) is known at this point. It may well be the case that these three concepts are actually chunked into one concept; however, the more conservative chunking hypothesis (of three concepts into two concepts) is used throughout.

In Table 4, the number of conceptual sets in working memory at the end of each of the three sentences is 1, 2, and 3 respectively, for a total of 6. The average number of concepts in memory across the three sentences is thus \(\frac{1+2+3}{3}=2.00\).

AVGINF
The average number of inferences made. This is the running sum of the number of inferences made divided by the number of sentences in the problem. Of interest here is the processing demand during bottom-up reading. A unit of inference is defined here as one of the five following possibilities: (a) establishing a relationship between two sets when that relationship is not explicitly described in the text; (b) instantiating an arithmetic action (e.g., JOIN) when the text does not mention a significant action; (c) determining the resulting set of an arithmetic action; (d) making a syntactic transformation to a more arithmetic-like form (e.g., "how many less than 5" to "5 less how many"); or (e) an unsuccessful referent search. Text integration and arithmetic action inferences are made when the conceptual representation
of a new sentence lacks the explicit information needed to establish text connections between conceptual sets or instantiate an appropriate arithmetic action. For example, in Table 4, an inference is made in the second sentence. Here, the simulation infers that the 3 cans that Jacob gave to Kathy are "part of his previously established set of 8." (An example sentence that would not have required this inference is "Jacob gave 3 of his cans to Kathy." In this case, the phrase "of his cans" would generate a direct link to the previous "set of 8," so a text integration inference would not be required.) In the example problem, the inference "the 3 given to Kathy are part of Jacob's original set of 8" is the only inference required. An inference is not needed to instantiate the arithmetic action SEPARATE... FROM as the parser has translated the action language "Jacob gave" into an explicit "transfer of possession from Jacob." The average number of inferences across the three sentences is thus \((0 + 1 + 0)/3 = 0.33\).

**AVGMEMINF**
This is the sum of the variables AVGMEM and AVGINF; that is, the combination of the average number of concepts in working memory and the average number of inferences that are made. This variable addresses the theory that children's total processing capacity is made up of (a) what they must remember as well as (b) what must be devoted to executing basic operations, such as making inferences (Baddeley & Hitch, 1974; Case, 1982; Just & Carpenter, 1992). In the context of arithmetic word problems, the fact that some problems require inferences may not be as critical as the number of concepts they must remember while they make those inferences.

**LOCAL Sentence-Level Variables**

**MAXMEM**
This is the maximum number of conceptual units in memory. This is the maximum number of conceptual units that appear in the simulation's working memory at the end of any one sentence. In the example of Table 4, the maximum number of conceptual units is 3, occurring in the last sentence.

**MAXINF**
This is the maximum number of inferences made during one sentence. In the example, the maximum number of inferences is 1, occurring in Sentence 2.

**MAXMEMINF**
This is the maximum sum of the number of concepts in working memory and the number of inferences that are made at the end of one sentence. This variable is needed in order to be sensitive to the peaks in total processing that occur while reading individual sentences. This measure is not necessarily the sum of MAXMEM and MAXINF, as those independent maximums
may not occur in the same sentence. In the example, MAXMEMINF is 3: Sentences 2 and 3 have equal sums of (2 + 1) and (3 + 0), respectively.

**Results and Discussion**

Global and local predictors were entered into separate multiple linear regression analyses. The three global predictor variables (Avg prefix) were entered into a regression equation with the probability of solution for a specific grade level as the dependent variable. For each grade level, a forward stepwise regression analysis was conducted.

Table 5 summarizes the results of the four regression analyses (grades K-3) using global variables. For each grade, the variables listed in Column 2 are those that met a .05 probability-to-enter and .10 probability-to-remove criterion in the stepwise regression. Column 3 presents the proportion of total variance accounted for by the global variable in that row for each grade. (Each row indicates a new grade level and thus an independent regression analysis.) For example, in grade K, AvgMemInf accounts for .493 of the variance. The last column presents the p values (two-tailed) associated with each of the regression coefficients.

Table 6 summarizes the results when the three local (sentence-level) variables are entered into the regression analyses. Considering the text integration processing at the individual sentence level, the combination of memory load and number of inferences (MaxMemInf) appear in the three youngest
grade levels. In grade K, MaxInf (maximum inferences) also appears as the second entry into the stepwise equation, although not significantly.

The combination of "concepts to remember" and "inferences made" as averaged across entire problems is clearly a significant predictor of young children's solution success, for both local and global measures. In particular, AvgMemInf accounts for at least 50% of the variance in children's solution probabilities in all four grade levels ($p < .001$). These results suggest that some problems are difficult because the task demands of producing a coherent and integrated representation exhaust a problem solver's cognitive resources. If the relationship between two or more sets is not clear, the problem solver must infer how these sets are related. Because the two (currently unrelated) sets occupy memory, the problem solver must use the remaining resources to work out how these sets are related, thereby increasing the load on working memory. If the requirements of integrating isolated sets into a structure that specifies their relationship exceed the available resources, the problem solver is left with a representation of one or more independent sets in memory.

If integrating isolated sets while keeping those sets in memory causes some problem solvers to exceed their cognitive resources, facilitating the integration of sets should decrease the demands on working memory. In the next three sections, we present results that show how our simulation is sensitive to varying demands on working memory caused by slight changes in problem wording. Because EDUCE is sensitive to slight changes in problem wording, the simulation can solve a number of reworded versions that have been shown to greatly affect solution success. The monitored performance of the simulation on these reworded versions can be entered into the equations derived from the previous regression analyses to obtain predictions on the relative impact of these rewordings. This includes rewordings that make problems both easier and harder than the original problem. The predictor equations from the Global regression analysis are shown in Table 7.

### Making Static Nonrelational Problems Easier

Combine problems are typical of a class of problems that do not contain significant actions (thus the term, static) and do not involve relational expressions such as "more than" (thus the term nonrelational). Table 8 shows
a "traditional" wording of this type of problem along with two potential rewordings. In all three versions, the first sentence describes a superset with a known amount. A subset with a known quantity is then described, followed by a question requesting the amount of the other subset.

The traditional wording has received considerable attention in the literature, mostly due to its level of difficulty. Considering all 18 problems in Riley and Greeno's (1988) benchmark set, this problem (Combine 5), on average, is the third most difficult problem. Children's success rates across the first four grades in the Riley et al. (1983; Riley & Greeno, 1988) studies reflect the difficulty: 22% for K, 33% for 1st, 55% for 2nd, and 75% for 3rd. These relatively low probabilities in even the higher grades are confirmed elsewhere (Davis-Dorsey et al., 1991; De Corte et al., 1985).

There are two potential rewordings that might help those students who are not able to solve this type of problem: (a) the use of language to facilitate text integration between sentences, thereby highlighting the relationship between the quantities (e.g., of them, the rest) and (b) those same changes along with the removal of the conjunction (and) and the word altogether.

The first potential alternative to the "traditional" wording is to include words that facilitate the process of text integration by highlighting the relationship between quantities. In Table 8, this is the reworded version that includes and, altogether, and of them.
As shown earlier, when EDUCE parses the second sentence of this reworded problem, the phrase "of them" generates an explicit reference to a previously mentioned set; that is, an explicit reference to David and Kathy's set of 8. This explicit reference identifies that David's 5 are part of the previously mentioned set of 8. In other words, EDUCE is sensitive to the "mathematics embedded in the natural language." On the other hand, in the second sentence of the traditional wording ("David has 5 soda cans"), no reference to the previous set of 8 is made in the text. Thus, EDUCE's representation of the second sentence of the traditional wording does not contain an explicit link back to the first sentence. When performing text integration, SELAH must make an inference to establish that David's 5 cans are part of David and Kathy's original set of 8 cans, as opposed to 5 totally unrelated cans.

Returning to the second problem in Table 8, the phrase, "The rest of them" in the third sentence marks Kathy's set as part of the 8 soda cans that are left as the result of a previous separation. In the traditional wording, no reference is made to Kathy's set, so SELAH must infer that Kathy's cans are part of the set of 8 as well as those remaining after separating out the cans belonging to David.

De Corte et al. (1985) and others have provided empirical support that this reworded version (and, altogether, and of them) significantly increases children's solution probability. In line with these results, our simulation predicts that including the phrases "of them" and "the rest" will increase solution probability. The average memory loads for these two problems are quite similar although the simulation performs three fewer inferences on the reworded version.

Table 8 compares first- and second-grade children's success on both the traditional and reworded versions as reported by De Corte et al. (1985) and predicted by the simulation. For both grades (first and second), the prediction is in line with De Corte's results, although with respect to De Corte's data, the simulation overestimates the difficulty of the traditional wording. For example, in first grade, De Corte et al. showed an increase in solution probability from 43% on the traditional wording to 57% on the reworded version, whereas the simulation predicted an increase from 28% to 59%. However, our estimation of difficulty on the traditional wording is in line with the first-grade results of Riley (33%) and Cummins (30%). It is not clear at this point whether the simulation is overestimating the types or number of inferences in the traditional wording or if the Belgian data reflect a difference in language use and/or instructional exposure to the traditional wording. Finally, as might be expected, the simulation's predictor equation for first grade is more sensitive to a change in the number of inferences than a later grade.

Another potential rewording of the traditional Combine 5 wording includes the phrases "of them" and "the rest" like the first rewording, yet the
conjunction (and) and the word altogether in the first sentence have been removed (see the bottom of Table 8). According to Cummins (1991), the critical difference between this rewording and the other rewording is the absence of the conjunction in the first sentence (the removal of altogether is considered secondary). More specifically, the conjunction of owners (David and Kathy) is often misinterpreted by young children as meaning each. Thus, in a sentence such as “David and Kathy have 8 soda cans altogether,” many children misinterpret this to mean: “David has 8 and Kathy has 8.” Such a misunderstanding would lead children to answer “8” when asked how many Kathy has. Cummins validated her hypothesis with results from a computer simulation, children’s recall protocols (Cummins et al., 1988), and the classification of solution error types (Cummins, 1991). In the latter study, children were found to commit given-number errors (i.e., the answer to the traditional wording is “8,” a number given in the problem) on 46% of the incorrect responses (see De Corte et al., 1985, for similar results).

Table 8 compares first-grade children’s success on both the traditional and reworded versions as reported by Cummins (1991) and De Corte et al. (1985) and predicted by the simulation.

Although our prediction is in line with that of Cummins, the explanations start from different points of reference. On the one hand, Cummins clearly showed that many children cannot map joint ownership of conjunctions and the term altogether (“David and Kathy have . . . altogether”) onto their working knowledge of part-whole relations. That is, rewording helps those children who misinterpret joint ownership as “each” because the unfamiliar language has been removed. On the other hand, our simulation highlights the difficulty of this problem even when children correctly interpret joint ownership and the term altogether. That is, even if children understand part-whole relations (Riley & Greeno, 1988) and even if they can map ambiguous words onto part-whole relations (Cummins, 1991), our simulation predicts that the traditional wording of this problem requires a relatively high demand on working memory. Rewording not only reduces the number of bridging inferences that must be made, but also eliminates the activation of potentially conflicting arithmetic actions.

More specifically, the simulation offers an information processing perspective as to why some children (approximately 26% in De Corte, Verschaffel, et al., 1985, and 27% of those who correctly interpreted altogether” in Cummins, 1991) use the wrong operation in their solution on the traditional wording (i.e., they add 8 and 5 rather than subtracting 5 from 8). While it is widely reported that children rely on surface clues for problem-solving hints (Burton, 1988; Clement, 1982; Langford, 1986;
NCTM, 1991, and others) the simulation offers new process-oriented explanations of comprehension "breakdowns" that lead to the application of keyword strategies. Many of the classic keywords associated with addition and subtraction problems imply arithmetic actions (e.g., altogether implies JOIN), but they do not always imply a one-to-one mapping to an arithmetic operation (e.g., altogether does not imply addition). Although JOINing intuitively suggests the adding of sets together, the instantiation of the action JOIN does not imply that one will use the addition operator. The distinction is subtle. Children (and teachers!) who do not appreciate the distinction often find themselves in disheartening situations. In the context of explaining wrong operation errors, the critical advantage of the Cummins' rewording is the removal of the word altogether. During simulation, the parser performs three (expert) tasks upon reading the word altogether: (a) partition the current set, although the qualifier that causes the partition (e.g., different owners, different colors) is currently unknown; (b) attempt to determine the qualifying dependency that partitions the set; and (c) instantiate the arithmetic action Join. For example, after the first sentence in the traditional wording is read, the simulation has the following representation in memory: (David & Kathy's 8 soda cans, JOIN). While the "sets which were joined" to form this set of 8 are still unknown, the critical point is that the Join action has been activated. Upon reading the second sentence of the traditional wording, the simulation infers that David's 5 are part of the previous set of 8. In order to construct David's 5 from the set of 8, the arithmetic action of Separating-From is needed. Thus, as the simulation reads the traditional wording, the Join action must be suppressed in favor of the Separating-From action. The simulation suggests that even for young children who do understand joint ownership, the instantiation of the Separating-From action may not be a sufficient condition to override the influence to Join. Children may rely on a keyword or "default to addition" strategy because they are unable to suppress previously activated information. In contrast, when altogether is removed from the initial sentence, the simulation does not instantiate Join and thus is not required to suppress it while processing the second sentence.

**Making Static Relational Problems Easier**

Static (no significant actions) relational (comparative language) problems are the most difficult addition and subtraction problems, as convincingly confirmed in empirical tests of children's solutions and recall protocols.

---

8 In the current simulation, instantiating an arithmetic action of Join is not analogous to picking the arithmetic operator of addition. The Join action simply means that the current set is partitioned and can be formed by joining two other (perhaps as yet unmentioned) sets. This is quite different than a "keyword" interpretation which maps the word "altogether" to the arithmetic operator of addition.
TABLE 9
Comparison of Predicted and Observed Static-Relational Compare 1 and Equalize Probabilities of Correct Solution for First Grade

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Example Problem</th>
<th>Simulation Prediction</th>
<th>Carpenter Data</th>
<th>Riley Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>Jacob has 2 soda cans. Chrissy has 8 soda cans. How many soda cans does Chrissy have more than Jacob?</td>
<td>0.48</td>
<td>—</td>
<td>0.28</td>
</tr>
<tr>
<td>Equalize</td>
<td>Jacob has 2 soda cans. Chrissy has 8 soda cans. How many soda cans does Jacob need to find to have as many as Chrissy?</td>
<td>0.90</td>
<td>0.91</td>
<td>—</td>
</tr>
</tbody>
</table>

(Cummins, 1991; Fuson, 1992; Lewis & Mayer, 1987; Stern, 1993), studies of children’s use of relational terms in the home (Walkerdine, 1990), and eye-movement experiments (Hegarty et al., 1992; Verschaffel et al., 1992). The traditional wording in Table 9 is typical of this class of problems, commonly referred to as Compare problems.

In the traditional wording, the cardinalities of two disjoint sets are compared. The question requests the difference between the sets with a focus on the larger set, that is, “how many more?” One possibility for making the traditional wording of this type of problem easier is to replace the static relational language (e.g., “more than”) with action language that describes a hypothetical act of making two sets equal (Carpenter et al., 1981, and others). With this “equalize” language, the static relational language (e.g., “how many more than”) in the traditional wording can be replaced with action language that describes a hypothetical act of making the two sets equal (e.g., “how many does one need to find to have as many as the other”). An equalize problem is shown in Table 9.

For all grade levels, the simulation predicts that the Equalize version will improve solution probability, especially for the youngest grades. This is due largely to the fact that the Equalize text provides an explicit action to Join (“need to find”) and an explicit reference to the result of that Join (“as many as Chrissy”). On the other hand, the simulation must make these two inferences when solving the situationally void traditional wording. Table 9 compares the solution probabilities of the traditional wording as found by Riley and Greeno (1988), the Equalize probabilities as found by Carpenter et al. (1981), and the simulation’s predicted solution probabilities for the traditional and Equalize wordings.
In addition to being sensitive to wording changes that make problems easier, the simulation also confirms empirical results for rewordings that cause some problems to be more difficult, for example, altering the time sequence of problems containing significant action language. As shown by Staub and Reusser (1995), altering the time sequence of a Change problem such that the sentences are not in strict chronological order makes the problem more difficult. The initial problem in Table 10 uses the traditional wording of a Change 1 problem. In the second (reworded) problem, a change to the presentational structure causes the first sentence to refer to the transfer of possession action and the second sentence to refer to the initial set that existed prior to the action. A critical difference between the traditional wording and the time-altered version is explained through the simulation when it solves these two problems. In the first sentence of the traditional wording, Alex is understood to possess 5 marbles and the simulation creates a set of 5 for Alex. The second sentence is understood to be a transfer to Alex, thus the 3 that are transferred are JOINed to his previous 5 as described by the “transfer-in” situation in the first two sentences. The JOIN action is explicit in the text and subsequent representation and need not be inferred.

In the time-altered version, the transfer of 3 to Alex occurs in the first sentence. Because Alex does not previously have any in his possession, the problem simulates the transfer as resulting in Alex’s set of 3 today. The second sentence indicates that Alex had 5 yesterday and the simulation makes another set for Alex with this amount. Because the program is simulating bottom-up, sentence-by-sentence reading, it does not “rerun” the first two sentences over in a chronological fashion. After interpreting the third sentence to mean a request for the number of marbles that Alex has now, the
COGNITIVE DEMANDS IN WORD PROBLEM SOLVING

387

simulation infers that time differentiates the three sets so it can JOIN Alex’s marbles-today and Alex’s marbles-yesterday. In short, when the simulation comprehends the time-altered problem, the transfer of 3 from Bethany to Alex in the first sentence does not instantiate a JOIN action. That action must be inferred based on the qualifying dependency of time (today, yesterday, now) at the end of the problem.

Staub and Reusser (1995) clearly showed that the time-altered version is more difficult to solve than the traditional version. Because the simulation is sensitive to the number of inferences that are required in each of these versions, the predictor equations verify their results. Table 10 presents children’s solution probability with the traditional wording (Riley & Greeno, 1988), children’s solution probability with the time-altered wording (Staub & Reusser, 1995), and the simulation’s predicted solution probabilities.

As expected, the simulation closely predicts the solution probability for the traditional wording.9 In addition, the simulation does predict a decrease in solution probability (70% success rate) but not as much of a decrease as Staub and Reusser found (63%). One possible reason may be that some children in the Staub study were confused by the difference between today and now, whereas the simulation was not. If this were the case, a number of children would be expected to give the answer of 3 (the number Alex had today). Staub reported, however, that only a small percentage of children who get it wrong give this answer (personal communication, April, 1994). Another possible reason from the cognitive simulation perspective is that one or both of the inferences required in the time-alteration version may be very difficult inferences for children to make. For example, the inference that marks the set (Alex has ? now) as the superset would appear to be a complex one; that is, the word now is the primary clue that the sets should be distinguished by time. Distinguishing superset and subsets by time is in fact a more abstract type of the more difficult Combine problems, where sets are typically distinguished by ownership (John has, Mary has, John and Mary have) or other features such as color (red marbles, blue marbles, red and blue marbles). Because the simulation counts all inferences with the same weight (i.e., +1), the simulation may be underestimating the difficulty of certain types of inferences, such as those due to semantic character (e.g., modality, degree of abstraction of objects and relationships). This is one indication that the types of inferences may be at least as important as, if not more important, than the number of inferences.

Before further discussion of these results, Simulation 2 is presented. This analysis explores the extent that reading processes at the sentence level might also account for some of the variance in solution probability.

* The simulation is expected to closely predict the solution probabilities on the traditional wordings because the predictor equations were derived from children’s solution probabilities when they solved problems with the traditional wordings. Note, however, that the equations are the result of an analysis using all three types of semantic structure, not just Change problems.
Simulation 2

By monitoring the amount of work it takes EDUCE to parse the individual sentences in each problem, this analysis attempts to isolate sentence-level reading measures that account for significant proportions of the variance in children's solution probability. As described earlier, EDUCE parses like an "expert" in this analysis; that is, working memory is unbounded and all words have a "correct" lexical entry.

Method

In this simulation, children's probability of solution was analyzed in regression analyses as a function of three predictor variables, where the predictor variables were sentence-level measures of the "amount of work" required to parse individual sentences into conceptual representations of their meaning.

Materials and Procedure

The materials are the same 18 benchmark problems as described in Simulation 1. For each problem, the simulation (EDUCE and SELAH together) read and completely solved the problem. In this case the focus of analysis is on the EDUCE component, whereas Simulation 1 focused on SELAH's contribution. While solving each problem, the English sentences in each of the 18 problems were input to the parser one at a time. For each sentence in each problem, the parser read and built a conceptual representation of that sentence. In this analysis, EDUCE printed out a record of its sentence-level measures. The output for all of the sentences in each problem was combined and analyzed to obtain scores for the variables described in the next section.

Predictor Variables

The probability of solutions obtained in the Riley et al. (1983; Riley & Greeno, 1988) studies were used in regression analyses as a function of three predictor variables. The variables represent computational measures of EDUCE's processes as it parses each sentence of each problem. A summary of the three variables is given here.

AVGPOST

This is the average number of expectations that are posted during the parsing of the sentences in a problem. EDUCE is a conceptual, "expectation-based" parser. Each word in the lexicon has a set of expectations or requests, where an expectation combines both syntactic and semantic information to "look" for certain concepts to occur elsewhere in the sentence. For example, the word gave posts a request to expect a human actor (semantic information) of the transfer to appear prior (syntax information) to the word gave. This variable is the total number of expectations that were posted for all sentences divided by the sum of the words in the problem. Note that posting
TABLE 11
Results of Regression Analysis for Simulation 2

<table>
<thead>
<tr>
<th>Grade</th>
<th>Variable</th>
<th>Variance Accounted For ($R^2$)</th>
<th>Regression Coefficient</th>
<th>Standard Error</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>AVGSTM</td>
<td>.146</td>
<td>0.92</td>
<td>0.48</td>
<td>.077</td>
</tr>
<tr>
<td></td>
<td>AVGSAT</td>
<td>.270</td>
<td>1.96</td>
<td>1.22</td>
<td>.131</td>
</tr>
<tr>
<td>1</td>
<td>AVGSAT</td>
<td>.181</td>
<td>2.94</td>
<td>1.56</td>
<td>.078</td>
</tr>
<tr>
<td>2</td>
<td>AVGSAT</td>
<td>.187</td>
<td>2.35</td>
<td>1.23</td>
<td>.073</td>
</tr>
<tr>
<td>3</td>
<td>AVGPOST</td>
<td>.173</td>
<td>0.22</td>
<td>0.12</td>
<td>.086</td>
</tr>
</tbody>
</table>

an expectation is not the same as satisfying an expectation (see AVGSAT). Some expectations may be posted but never satisfied.

**AVGSAT**
This is the average number of expectations that become satisfied in a problem. As each word is processed, outstanding expectations are checked one at a time to see if an expectation is satisfied. This variable is the total number of expectations that become satisfied divided by the number of times the parser checks all the outstanding requests that are not yet satisfied. In general, for each sentence, the parser checks all outstanding requests once for each word and once at the end of the sentence, so the number of times the parser checks the requests for each sentence is typically the sum of the number of words in each sentence plus one.

**AVGSTM**
This is the average number of concepts in short-term memory (STM) during the parse of a sentence. As EDUCE parses a sentence, concepts are stored in STM (e.g., a human named Kathy, a state of possession, a soda can object) and eventually become merged into an overall conceptual representation as expectations become satisfied and concepts are “chunked” together (e.g., [Kathy] < POSSESSES> [soda can object]). The parser monitors the total number of concepts that entered working memory for each sentence. This variable represents the total number of concepts that existed in working memory for all sentences in a problem divided by the sum of the words in the problem.

**Results and Discussion**
To investigate the effects of the variables, three predictors were entered into a multiple linear regression analysis with the probability of solution for a specific grade level as the dependent variable. These variables are described in the Method section.

---

10 The final scores for the three sentence-level variables for all 18 problems in Simulation 2 as well as the scores for the three global variables in Simulation 1 are available on request.
Table 11 summarizes the results of the four analyses (one for each grade) for children in kindergarten through third grade. Because no variable reached a $p = .05$ value, Table 11 reflects the entries when a liberal $p = .15$ probability to enter was used. Even with this liberal probability to enter value, parsing variables account for less than 20% of the variance in all four grades.

Children's solution probabilities do not vary with the "amount of conceptual work" at the sentence level as implemented by EDUCE because "expert" parsing of individual sentences in traditionally difficult problems does not involve significantly more work than parsing sentences in the easier problems. More specifically, the measures used in EDUCE simply do not capture the semantic/conceptual difficulties involved; that is, they do not capture the difficulty in conceptualizing the relations involved, relations that in other than Compare problems are usually distributed among several (more than one) sentences and are more relevant to the processes of text integration. Clearly, semantic understanding is involved in parsing in order to end up with a semantic representation, where the semantics in this case involves mathematics. However, EDUCE's parsing (or parsing measures) deals with the arrangement of symbols that represent the situation (and not with their semantic content). In general, parsing effort depends on both the complexity of structures at the symbolic level (i.e., the complexity of syntax), and the deviation of these syntactic structures from (what we imagine to be) the unique semantic/conceptual/math representation of the sentence (i.e., the extent to which word groupings deviate from the semantic/conceptual grouping presumably in our heads). For example, article-adjective-noun ("a red ball") will be easy to parse because words representing the attributes of a concept are located close to the concept, as they presumably are semantically in our heads (and because this syntax has often been encountered by young children in reference to the everyday world). This idea can be represented roughly as difficulty = syntactic complexity divided by efficiency of the syntax, where syntactic complexity may reflect the semantics of the sentence (a semantically complex sentence may require corresponding syntactic complexity), but efficiency is relative only to a given semantic representation. This is consistent with the pretensions of EDUCE. A reason for the discrepancy, or noncorrelation of EDUCE's results with the data, may be that understanding a sentence also depends on the ability to come out with any structure at all—one has to understand comparatives in order to understand comparative sentences, but EDUCE (parsing in an "expert" mode) does not measure this ability. Rather, the results "show" that if that kind of understanding is there, then (apparently) expectations and STM load do not make much difference for the problems given.

If one assumes a relationship between "amount of processing" and time of solution, the "negative" result of this analysis is in line with eye movement results from Verschaffel et al. (1992) and Hegarty et al. (1992), which
show that with college-level students, the time for an initial reading of the problem (the translation phase) does not significantly vary between problems that differ in probability of solution.\(^{11}\) With respect to our simulation, Hegarty's translation phase is analogous to EDUCE's parsing of all the individual sentences. Hegarty's eye movement results show that neither high-accuracy (one or no wrong answer on four test problems) nor low-accuracy college-level students spend more time on the initial reading of hard Compare problems than they spend on the initial reading of easy Compare problems. However, students did spend more time in the integration and planning stages on the more difficult problems. Thus, eye movement data show that difficult problems require additional integration but not longer (initial) reading times. This result is consistent with the relative proportions of variance accounted for in these exploratory analyses. That is, the amount of parsing effort as currently measured by EDUCE does not vary with solution probability, whereas SELAH's current measures of the load on working memory during text integration do vary with solution success.

The implications of this negative result are mixed. On the one hand, neither EDUCE's measures of expert parsing nor the initial reading times of Hegarty's high-accuracy students vary with problem difficulty. Because these students are able to solve the problems correctly, they presumably can generate correct interpretations of the individual sentences. As discussed earlier, one of our processing assumptions is that EDUCE is currently parsing with sufficient knowledge to interpret all the sentences. Parsing in this "expert" mode, the results of this simulation extend previous results to a new measure, namely, the amount of parsing effort rather than the amount of time to read (Hegarty et al., 1992) or error rate (Lewis & Mayer, 1987). On the other hand, it is equally possible that significant differences do exist at the individual sentence level but EDUCE is neither simulating nor monitoring the "critical" processes. For example, EDUCE does not simulate the richness of the eye movement data (e.g., people make multiple regressions to previously read lines of the problems after their initial reading), nor does Simulation 2 consider the storage demands from previous sentences as it parses a new sentence. We return to this second point in the general discussion.

**GENERAL DISCUSSION**

At the beginning of this article, we presented what we consider to be a benchmark of phenomena to account for in a simulation of word problem solving (see Table 2). In the following discussion, we present how the EDUCE/

\(^{11}\) Hegarty et al.'s (1992) results only address Compare problems.
SELAH simulation satisfies or fails to satisfy the criteria in this benchmark set. Our theoretical objective is to extend the linguistic development view by proposing how changes in natural language affect the construction of mathematical connections in working memory. The educational objectives are to increase teacher awareness of the multiple sources of problem difficulty and to show how slight changes in problem wording affect a choice of the "next best problem" to present.

Semantic Structure and Problem Wording
Regarding semantic structure, the current bottom-up simulation successfully solves all 18 problems in the standard categorization of Change, Combine, and Compare. EDUCE successfully parses all sentences in these problems and SELAH successfully maps the connections made during text integration onto an action-based representation, including those problems that do not contain significant action language (Combine and Compare). Similar to the results of previous simulations, the regression analyses presented earlier clearly show a good fit of our simulation with the data on children's success on these standard 18 problems as well as other variants such as Equalize problems and various rewordings, for example, changes in chronological order or the presence of explicit set references.

Despite the overall good fit, the simulation overestimates the difficulty of Combine 5 & 6 for the youngest problem solvers (grades K and 1). Combine 6 and 5 have the first and second highest AvgMemInf values. However, the Riley et al. (1983; Riley & Greeno, 1988) data show that for children in kindergarten and first grade, these two problems are closer to the median when all 18 problems are ranked according to problem difficulty.

One explanation may be that the simulation's deterministic manner of making all possible inferences at a given point in time is not a sufficient portrayal of how very young children solve problems. For example, in the initial sentence of a Combine 5 or 6 problem, "Shelly and Fred have 8 floppy disks altogether," the word altogether triggers the simulation to make the inference that an arithmetic action of Join on two sets must have occurred to make this set of eight. Yet this particular inference is clearly associated with a more "expert-like" understanding of the word altogether. We view this particular overestimation as a consequence of our initial processing assumption that the simulation model the overall demands of bottom-up processing. Said differently, the simulation's measures for Combine 5 and 6 state that if the problem solver "correctly" interprets sentences containing ambiguous phrases, then a determination of the relationships in the problem requires a relatively high load on working memory. However, it is clear from the literature that many young children experience difficulty at the stage of comprehending sentences; for example, they can barely produce coherent recalls or drawings of sentences that contain comparatives (Cummins,
If one assumes that older children are less likely to have difficulty with understanding ambiguous phrases, then problem difficulty may be more closely associated with the overall demands of arriving at an integrated representation of the situation. This explanation is supported by the result that older children (second and third grade) do find the Combine 5 and 6 problems difficult relative to the other problems: These two problems tie for the third most difficult problem for second graders and are the second and third most difficult problems for third graders.

General Context and Affective Factors
Word problems embedded into a familiar setting are much easier to solve than those that must be solved without such support. Familiar settings that have been shown to facilitate solution success include the presence of familiar objects and/or names (Davis-Dorsey et al., 1991; Staub & Reusser, 1995). As mentioned earlier, our simulation is not sensitive to these factors, nor is it entirely clear how a system might account for affective factors such as the use of familiar names. However, Kintsch’s (1988) construction–integration model, a combination of symbolic and connectionist processes, does suggest a way to approximate the elaboration and integration of situational and arithmetic information that is needed in order to account for general context effects.

In regards to the text integration component, SELAH and the construction–integration model of Kintsch (1988) perform the process of text integration at sentence boundaries (i.e., at the end of a sentence). SELAH deals specifically with bridging inferences (Haviland & Clark, 1974); that is, the integration of sentences when no explicit reference is available, and the demands that these inferences place on working memory. On the other hand, Kintsch’s integration phase deals with linguistic input as well as input constructed from the comprehender’s (real-world) knowledge base in a connectionist manner (e.g., Rumelhart & McClelland, 1986). In the construction phase prior to integration, propositions from the current sentence in the original text activate propositions from a general knowledge network of propositions and contribute to a network of potentially conflicting propositions to represent the situation. In the integration phase, activation is spread around the network of propositions until the system stabilizes, that is, with respect to solving arithmetic word problems, multiple part–whole schema compete for propositions that satisfy their constraints. The ability to integrate real-world knowledge of situations with the arithmetic knowledge of part–whole roles “is the aspect that the (Kintsch) model deals with most effectively” (Kintsch, 1988, p. 178). Our simulation’s current inability to activate semantically related knowledge means that it is unable to account for situationally rich context effects.
The Locus of Problem Difficulty

The sentence-level variables account for small proportions of the variance in the youngest children's solution probabilities in Simulation 2, but text integration variables are clearly the strongest predictors of solution success, as shown in Simulation 1. Yet without independent evidence for our process model, we are not currently in a position to conclude from these exploratory analyses that children's difficulties are due to the demands of text integration but not to the processes of sentence-level reading. In fact, the small amounts of variance accounted for by the sentence-level variables may be more a function of the "fixed" level of expertise and/or the current selection of variables than the fact that nothing of interest is occurring at this level of comprehension. In particular, the simulation's ability to focus at fine-grained levels of processing is revealing tacit factors that may contribute to children's difficulties with relational language and sentence-level processing in general. These factors are discussed in the context of suggesting why children are often unable to map "more than" onto their part-whole knowledge.

As discussed earlier, sentences containing the relational expression more than indicate a specific referent to a previous amount. A review of how the simulation handles the second sentence of the following Compare problem suggests new reasons why many children interpret relational expressions as assignment statements, as shown by Cummins et al. (1988; Cummins, 1991).

Betsy has 6 bottles. Bill has 3 bottles more than Betsy.
How many bottles does Bill have?

EDUCE currently interprets the second sentence to mean that "Bill has 3 bottles in addition to [the number of bottles that Betsy has]." Of particular interest is the processing that is involved in the middle of reading this sentence. Having read the first four words ("Bill has 3 bottles"), EDUCE represents this to mean that Bill is in possession of 3 bottles. The next word, more, instantiates an in-addition-to interpretation with the last two words, "than Betsy," supplying the referent. Relational expressions such as "more than Betsy" require EDUCE to handle ellipsis, that is the omission of a number of words from a phrase that are expected to be understood. In this case, EDUCE interprets the phrase "more than Betsy" as "more than Betsy (has bottles)." The words in parentheses (has bottles) represent the ellipsis. An inability to handle this ellipsis means that the referent is not the "set possessed by Betsy," but simply "Betsy." In other words, EDUCE would

12 It is of interest that this interpretation is correct when one "has more," but it is not correct when one "has less." That is, if I have 3 more than you, I am in possession of the 3. However, if I have 3 less than you, I am not in possession of the 3.
interpret the sentence to mean “Bill has 3 bottles in addition to Betsy,” a (mis)interpretation very similar to that reported in children’s recall protocols (Cummins et al., 1988) and drawings (Cummins, 1991). We agree with Cummins that children misinterpret comparatives because they are not able to map these forms onto their knowledge of logical set relations, but we suggest that an inability to make this mapping is due in part to (a) children’s reliance on the qualitative meaning of “more” and (b) children’s lack of resources needed to process the end of the sentence (e.g., the ellipsis) and thereby augment the current interpretation that “Bill has 3 bottles.” The second suggestion is of particular interest, especially in light of Just and Carpenter’s (1992) view that readers with higher capacity are able to maintain multiple interpretations of an ambiguous sentence, whereas low-capacity readers resort to a single interpretation scheme. In the context of processing the second sentence of this problem, young readers simply may not have the capacity to process the end of the sentence. It is in regard to these issues that EDUCE’s capability to simulate the construction of meanings allows for new hypotheses concerning the locus of problem difficulty.

Returning to the explanation of the simulation of this problem, SELAH performs a referential search and makes two inferences after EDUCE interprets the second sentence to mean: “Bill is in possession of 3 bottles in addition to [those Betsy has].” More specifically, SELAH first searches memory for a “set possessed by Betsy.” In this case, the search is successful and SELAH augments the representation to mean: “Bill is in possession of 3 bottles in addition to [the amount of bottles that Betsy has].” Because EDUCE did not encounter any significant actions in the text, SELAH infers the arithmetic action, Join. In addition to “Bill’s 3 bottles” explicitly described in the text, SELAH also infers that Bill is also in possession of the same number of bottles as in Betsy’s possession. An inability to make this second inference is another potential reason why children may resort to the assignment interpretation; that is, they may misunderstand it to mean “Bill is in possession of 3 bottles. (Betsy has bottles).”

Demands on Working Memory

A significant result from our simulation is that a combined measure of the number of concepts to remember and the number of inferences to make

---

13 The most complex Compare problems involve a referential search that fails at this point. For example, given the following sentences: “Betsy has 6 bottles. Betsy has 2 bottles more than Bill,” EDUCE interprets the second sentence to mean “Betsy has 2 bottles in addition to [those Bill has].” The failed referential search occurs when SELAH searches memory for a set of bottles possessed by Bill. As SELAH cannot find any previously mentioned sets associated with Bill, it must (a) make the inference that Bill is indeed in possession of a set of bottles, although the amount of that set is unknown or (b) SELAH must transform the relation in the second sentence to “Bill has 2 bottles less than Betsy,” as suggested by Lewis and Mayer (1987).
while remembering those concepts is a consistent predictor of children's problem-solving success in seven of the eight grade analyses: A global measure averaged across the entire problem (AvgMemInf) is significant across all four grade levels (K-3) and a local measure (MaxMemInf) appears in the three youngest grades. If one considers a child's maximum level of activation as fixed with storage and processing competing for activation (as suggested by Baddeley & Hitch, 1974; Case, 1982; Just & Carpenter, 1992, and others), a word problem that requires a relatively large number of concepts to be held in memory also limits the resources that can be devoted to executing basic operations, including the making of inferences. The predictor "average of memory and inferences" (AvgMemInf) offers new directions for testing storage/processing trade-offs during word problem solving. In this way, the simulation extends Kintsch and Greeno's (1985) suggestions on processing requirements to measures of "storage" and "work" that are sensitive to changes in problem wording.

In the context of our bottom-up "expert," the inferences necessary for text integration are critical. An ability to arrive at a correct arithmetic operator is directly dependent on establishing the relations between sets, which in turn is dependent on an integrated representation of the text. Unlike narrative texts, word problems tend to be very brief and consecutive sentences contain limited degrees of overlap (cf. Haviland & Clark, 1974). Children who are unable to make the inferences that lead to "mathematical connections" are confronted with independent sets in memory and must resort to ad hoc strategies (e.g., a keyword or first-number-given strategy) as confirmed in the literature (Cummins et al., 1988; De Corte et al., 1985; and others).

In summary, children's difficulties with arithmetic word problems may be due in part to an inability to make text integration inferences, especially when a relatively high number of concepts occupy memory. The implications for instruction are twofold. First, the processes of linguistic and mathematical comprehension are tightly coupled in arithmetic word problem solutions; however, there are fine-grained methods of altering a problem's probability of solution in each area. Because making "mathematical connections" is so critical, this research suggests that rewording problems in ways that imply or even explicitly state the relationships between sets is a critical step toward helping those students who cannot yet make the necessary inferences. This approach is particularly important insofar as children are often expected to make complex text integration inferences that lead to mathematical connections while they are just beginning to read. The second educational implication, related to the first, is that a finer grained classification of word problems is emerging. Based on the results of the regression analyses, the simulation is able to dynamically predict the difficulty of a problem based on some of the processes in text comprehension. The goal is
to develop a sequence of problems beyond the traditional classifications of semantic structure, in particular, a classification in which it is possible for the natural language expressions to vary while the mathematical content remains fixed. There are two purposes for such a classification. First, the sequencing of problems must be sensitive to the fact that a slight change in wording can have a significant impact on solution success. Determining the level of difficulty of a problem is a function of many factors, including the processes of text integration that vary with problem wording. This has direct implications for teachers who generate their own handouts, as well as for the writers of textbooks and achievement tests, especially given the new emphasis on exposing children to more "realistic" and challenging two-step problems. Second and most important, we view children's mastery or ownership of natural language that embeds mathematical concepts as a critical measure of obtaining competence in mathematics. This view defines a profitable learning trajectory as one that continually exposes children to the rich collection of mathematical expressions that are needed to verbalize and write about mathematical situations and the quantitative relationships in those situations.

**FUTURE DIRECTIONS**

The ability to explain problem difficulty rankings in terms of the processing of particular wordings brings us closer to a more comprehensive model of arithmetic word problem solving processes. The present work focuses on text integration processes and suggests that, even within this specific domain, the interaction of two tasks (e.g., concepts to remember and inferences made) may cause difficulty, where each individual task may not. Similarly, at a higher level, it seems clear that language comprehension processes and their relation to mathematical processes should be included in any model of arithmetic word problem solving, because consideration of mathematical or linguistic processes alone do not reflect all that is going on in the process of word problem solution.

A related reason for studying linguistic and text integration processes along with mathematic processes is the determination of possible relationships or distinctions between the various developmentally determined competences that underlie each process. From the perspective of the development of number concepts, for example, Case (1985) postulated successive conceptual structurings, built up recursively from preceding structures, which are necessary for a child to understand the mathematics of arithmetic word problems with differing semantic structures. Some linguistic competences could apparently be related to some of these structurings. An understanding of nonquantitative verbal comparisons, for example, may be a prerequisite for an understanding of quantitative comparisons and may therefore refer
to a similar conceptual structure. In the case of mathematically significant
natural language, the determination of mathematical connections in EDUCE/
SELAH in effect presumes the mediation of elementary competences or
structurings common to both linguistic and mathematical expression. By
analyzing semantic structure further, we can arrive at possible competences,
which may then be found to play a role in other difficulties as well. Other
aspects of natural language processing, (e.g., the additional memory load
imposed by an “unnatural” or inconsistent ordering of sentences or sentence
components) are presentation- rather than mathematics-related and may
therefore require different competences, in addition to the ability to handle
both textual and mathematical processing at the same time, as discussed
earlier. The described research provides the future possibility to integrate
competences for both tasks into a developmental theory that relates stages of
mathematical understanding with stages of natural language understanding.

APPENDIX A: SUMMARY OF SIMULATION’S
MEMORY LOAD AND INFERENCES

This appendix summarizes the text integration scores used in Simulations 1
and 2 for each of the 18 problems in the benchmark set. A sample of prob-
lem wordings is shown in Table 12.

Specifically, for each problem, a summary is given that explains how the
simulation arrives at scores for AVGMEM (the average number of units
held in working memory) and AVGINF (the average number of inferences
made). In addition, the summaries give an account of the types of infer-
ences that are made. See LeBlanc (1993a) for more detail.

AVGMEM
This is the average number of conceptual units in memory. This is the running
sum of the number of conceptual units that appear in the simulation’s
working memory at the end of each sentence divided by the number of sen-
tences in the problem. In these simulations, the working memory was un-
bounded, as opposed to setting it at some predetermined size (cf. Fletcher,
1986; Just & Carpenter, 1992). A conceptual unit is defined as either:

1. A single conceptual set (e.g., [Jacob’s 8]).
2. An arithmetic action (e.g., JOIN or SEPARATE_FROM).
3. An action and conceptual set(s) that have been “chunked” together.

In general, each set is counted as one unit and an arithmetic action is
counted as one unit, unless the two sets that are involved in the action both
have known amounts. If both sets have known quantities, the three total
units (two sets and one action) are “chunked” or merged into two units.
For example, [JOIN 3 & 5] is counted as two units: ([JOIN 3] [TO 5]),
whereas [JOIN 3 & some] is counted as three units: ([JOIN] [3] [TO some]).
TABLE 12
Sample Problem Wordings Used in Simulations 1 and 2

| Change 1 | Jacob had 5 soda cans. Then Kathy gave Jacob 3 soda cans. How many soda cans does Jacob have now? |
| Change 2 | Jacob had 8 soda cans. Then he have 3 soda cans to Kathy. How many soda cans does he have now? |
| Change 3 | Jacob had 5 soda cans. Then Kathy gave Jacob some soda cans. Now Jacob has 8 soda cans. How many soda cans did Kathy give Jacob? |
| Change 4 | Jacob had 8 soda cans. Then he gave some soda cans to Kathy. Now Jacob has 3 soda cans. How many soda cans did Jacob give to Kathy? |
| Change 5 | Jacob had some soda cans. Then Kathy have Jacob 5 soda cans. Now Jacob has 8 soda cans. How many soda cans did Jacob have in the beginning? |
| Change 6 | Jacob has some soda cans. Then he gave 5 soda cans to Kathy. Now Jacob has 3 soda cans. How many soda cans did Jacob have in the beginning? |

| Combine 1 | Jacob has 5 soda cans. Kathy has 3 soda cans. How many soda cans do they have altogether? |
| Combine 2 | Jacob and Kathy have some soda cans. Jacob has 5 soda cans. Kathy has 3 soda cans. How many soda cans do they have altogether? |
| Combine 3 | Jacob has 5 soda cans. Kathy has some soda cans. Jacob and Kathy have 8 soda cans altogether. How many soda cans does Kathy have? |
| Combine 4 | Jacob has some soda cans. Kathy has 3 soda cans. Jacob and Kathy have 8 soda cans altogether. How many soda cans does Jacob have? |
| Combine 5 | Jacob and Kathy have 8 soda cans altogether. Jacob has 5 soda cans. How many soda cans does Kathy have? |
| Combine 6 | Jacob and Kathy have 8 soda cans altogether. Kathy has some soda cans. Jacob has 5 soda cans. How many soda cans does Kathy have? |

| Compare 1 | Jacob has 8 soda cans. Kathy has 3 soda cans. How many soda cans does Jacob have more than Kathy? |
| Compare 2 | Jacob has 8 soda cans. Kathy has 3 soda cans. How many soda cans does Kathy have less than Jacob? |
| Compare 3 | Jacob has 3 soda cans. Kathy has 2 soda cans more than Jacob. How many soda cans does Kathy have? |
| Compare 4 | Jacob has 5 soda cans. Kathy has 2 soda cans less than Jacob. How many soda cans does Kathy have? |
| Compare 5 | Jacob has 8 soda cans. Jacob has 5 soda cans more than Kathy. How many soda cans does Kathy have? |
| Compare 6 | Jacob has 8 soda cans. Jacob has 3 soda cans less than Kathy. How many soda cans does Kathy have? |

The rationale for reducing the memory load through chunking is based on the fact that in the situation with two known quantities, all the necessary information for a solution is available. In short, because SELAH relies on mental representation of sets and actions, the assumption is that the cognitive demands of joining two sets with known quantities in one’s head is slightly less than joining two sets where one set has an unknown quantity.
AVGINF
This is the average number of inferences made. This is the running sum of the number of inferences that are made divided by the number of sentences in the problem. Of interest here is the processing demand during bottom-up reading. A unit of inference is defined here as one of the five following possibilities:

1. Establishing a relationship between two sets when that relationship is not explicitly described in the text.
2. Instantiating an arithmetic action (e.g., JOIN) when the text does not mention a significant action.
3. Determining the resulting set of an arithmetic action.
4. Making a syntactic transformation to a more arithmetic-like form (e.g., "how many less than 5" to "5 less how many").
5. An unsuccessful referent search.

Text integration and arithmetic action inferences are made when the conceptual representation of a new sentence (as produced by the parser, EDUCE) lacks the explicit information needed to establish text connections between conceptual sets or instantiate an appropriate arithmetic action.

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Sentence</th>
<th>AvgInf</th>
<th>AvgMem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Change 1</td>
<td>make set</td>
<td>transfer-in: explicit result of JOIN</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[JOIN J’s 3] [JOIN J’s 3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[TO J’s 5] [TO J’s 5]</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RE J’s ?]</td>
<td></td>
</tr>
<tr>
<td>Change 2</td>
<td>make set</td>
<td>transfer-out: explicit result of SP-FR</td>
<td>0.33</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[SP K’s FR] [SP K’s FR]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[J’s 8] [J’s 8]</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RE J’s ?]</td>
<td></td>
</tr>
<tr>
<td>Change 3</td>
<td>make set</td>
<td>transfer-in: explicit result of JOIN Kathy gave ?</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[JOIN] [JOIN]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[J’s some] [J’s some]</td>
<td>3.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[TO J’s 5] [TO J’s 5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RE J’s 8] [RE J’s 8]</td>
<td></td>
</tr>
<tr>
<td>Problem Type</td>
<td>Sentence</td>
<td>AvgInf</td>
<td>AvgMem</td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td><strong>Change 4</strong></td>
<td>make set</td>
<td>transfer-out: <em>Infer, &quot;some to K&quot; part-of 8; explicit SP-FR</em></td>
<td>[J's 8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[K's some FR]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[J's 8]</td>
</tr>
<tr>
<td><strong>Change 5</strong></td>
<td>make set</td>
<td>transfer-in: <em>explicit JOIN of JOIN</em></td>
<td>[J's some]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[TO J's some]</td>
</tr>
<tr>
<td><strong>Change 6</strong></td>
<td>make set</td>
<td>transfer-out: <em>Infer, &quot;5 to K&quot; part-of some; explicit SP-FR</em></td>
<td>[J's some]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Combine 1</strong></td>
<td>make set</td>
<td>make set</td>
<td>infer, JOIN: <em>They have?</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Combine 2</strong></td>
<td>make set</td>
<td>make set</td>
<td>infer, JOIN: <em>They have?</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Combine 3</strong></td>
<td>make set</td>
<td>make set</td>
<td>infer, JOIN: <em>K has?</em></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continued)
<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Sentence</th>
<th>AvgInf</th>
<th>AvgMem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Combine 4</strong></td>
<td>make set</td>
<td>make set</td>
<td>J has ?</td>
</tr>
<tr>
<td></td>
<td>make set</td>
<td>Infer, JOIN; Infer, some part of 8 Infer, 3 part of 8</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[J's some]</td>
<td>[J's some]</td>
<td>[JOIN]</td>
</tr>
<tr>
<td></td>
<td>[K's 3]</td>
<td>[K's 3]</td>
<td>[TO J's some]</td>
</tr>
<tr>
<td></td>
<td>[SEP J's 5]</td>
<td>[SEP J's 5]</td>
<td>[FR J&amp;K's 8]</td>
</tr>
<tr>
<td></td>
<td>[FR J&amp;K's 8]</td>
<td>[RE K's ?]</td>
<td>[SEP J's 5]</td>
</tr>
<tr>
<td></td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J's 5]</td>
</tr>
<tr>
<td><strong>Combine 5</strong></td>
<td>make set; Infer, JOIN</td>
<td>make set; Infer, 5 part of 8 suppress JOIN; Infer, SEP-FR</td>
<td>make set - K's ?</td>
</tr>
<tr>
<td></td>
<td>J has ?</td>
<td>Infer, ? part of 8 Infer, ? is the result of SP-FR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[J &amp; K's 8]</td>
<td>[SEP J's 5]</td>
<td>[SEP J's 5]</td>
</tr>
<tr>
<td></td>
<td>[JOIN]</td>
<td>[FR J&amp;K's 8]</td>
<td>[RE K's ?]</td>
</tr>
<tr>
<td></td>
<td>[SEP J's 5]</td>
<td>[FR J&amp;K's 8]</td>
<td>[SEP J's 5]</td>
</tr>
<tr>
<td></td>
<td>[FR J&amp;K's 8]</td>
<td>[RE K's ?]</td>
<td>[SEP J's 5]</td>
</tr>
<tr>
<td></td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J's 5]</td>
</tr>
<tr>
<td><strong>Combine 6</strong></td>
<td>make set; Infer, JOIN</td>
<td>make set; Infer, some part of 8 suppress JOIN; Infer, SEP-FR</td>
<td>make set; K has ?</td>
</tr>
<tr>
<td></td>
<td>J has ?</td>
<td>Infer, part of 8 Infer, 5 is the result of SP-FR</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[J &amp; K's 8]</td>
<td>[SEP]</td>
<td>[SEP]</td>
</tr>
<tr>
<td></td>
<td>[JOIN]</td>
<td>[K's some]</td>
<td>[K's ?]</td>
</tr>
<tr>
<td></td>
<td>[SEP J's 5]</td>
<td>[FR J&amp;K's 8]</td>
<td>[FR J&amp;K's 8]</td>
</tr>
<tr>
<td></td>
<td>[FR J&amp;K's 8]</td>
<td>[RE K's ?]</td>
<td>[SEP J's 5]</td>
</tr>
<tr>
<td></td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J&amp;K's 8]</td>
<td>[RE J's 5]</td>
</tr>
<tr>
<td><strong>Compare 1</strong></td>
<td>make set</td>
<td>make set</td>
<td>referent found</td>
</tr>
<tr>
<td></td>
<td>J has ?</td>
<td>Infer, JOIN; Infer, J has an amount equal to K's 3; Infer, set of 8 is result of action</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[J's 8]</td>
<td>[J's 8]</td>
<td>[JOIN]</td>
</tr>
<tr>
<td></td>
<td>[K's 3]</td>
<td>[J's ?]</td>
<td>[TO (J's 3)]</td>
</tr>
<tr>
<td></td>
<td>[RE J's 8]</td>
<td>[RE J's 8]</td>
<td>[RE J's 5]</td>
</tr>
<tr>
<td><strong>Compare 2</strong></td>
<td>make set</td>
<td>make set</td>
<td>(K has qty = ? less than J); referent found: Infer, SEP-FR;</td>
</tr>
<tr>
<td>Problem Type</td>
<td>Sentence</td>
<td>AvgInf</td>
<td>AvgMem</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>Compare 3</td>
<td>make set [K’s 2]</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>referent found;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, JOIN;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, K also has</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>amt = to J’s 3;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, K’s total is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RF of action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compare 4</td>
<td>make set [K’s 2]</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td></td>
<td>referent found;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, SEP-FR;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, “2 less 5” to “5 less 2”;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, K’s set is</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>RE of action</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compare 5</td>
<td>make set [J’s 5];</td>
<td>1.33</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, failed ref. search;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, set for K;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, JOIN;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Infer, J’s 8 is RE of action</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continued)
### Table 13 (Continued)

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Sentence</th>
<th>AvgInf</th>
<th>AvgMem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compare 6</td>
<td>make set K?</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(J has qty= 3 less than K's)</td>
<td></td>
<td>1.33</td>
</tr>
<tr>
<td></td>
<td>Infer, failed ref. search: Infer, SEP-FR: Infer, &quot;3 less K&quot; to &quot;K's less 3&quot;; Infer: J's 5 is RE of action</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[J's 5] [SEP] [SEP] [3] [3] [FR K's ?] [FR K's ?] [RE J's 5] [RE J's 5]</td>
<td></td>
<td>3.00</td>
</tr>
</tbody>
</table>

### References


