Spring 1993

A computer model of the role of text integration in the solution of arithmetic word problems

Mark David LeBlanc
University of New Hampshire, Durham

Follow this and additional works at: https://scholars.unh.edu/dissertation

Recommended Citation
https://scholars.unh.edu/dissertation/1732

This Dissertation is brought to you for free and open access by the Student Scholarship at University of New Hampshire Scholars' Repository. It has been accepted for inclusion in Doctoral Dissertations by an authorized administrator of University of New Hampshire Scholars' Repository. For more information, please contact nicole.hentz@unh.edu.
A computer model of the role of text integration in the solution of arithmetic word problems

Abstract
Understanding arithmetic word problems involves a complex interaction of text comprehension and mathematical processes. This work presents a computer model of the hypothesized processes that are required of a young student solving arithmetic word problems, including the processes of sentence-level leading and text integration. Unlike previous computer simulations of word problem solving, which neglect the early stages of text processing, this model forces a detailed consideration of the linguistic process, which is being increasingly recognized as a primary source of difficulty. Three experiments were conducted to isolate critical test comprehension processes. Children's probability of solution was analyzed in regression analyses as a function of the model's text comprehension processes. A variable measuring the combined effects of the load on working memory and text integration inferences accounted for a significant amount of variance across four grade levels (K-3). The results suggest new process-oriented measures of determining why a particular word problem may be difficult, especially for young students. An implication for education is the potential for a difficulty-differentiated network of problems that includes a multiple number of rewordings for each "traditional" problem wording as an aid for classroom assessment, curriculum development and future computer-based learning environments.

Keywords
Computer Science, Education, Elementary, Engineering, System Science

This dissertation is available at University of New Hampshire Scholars' Repository: https://scholars.unh.edu/dissertation/1732
INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313-761-4700 800-521-0600
A computer model of the role of text integration in the solution of arithmetic word problems

LeBlanc, Mark David, Ph.D.
University of New Hampshire, 1993
A COMPUTER MODEL OF THE ROLE OF TEXT INTEGRATION
IN THE SOLUTION OF ARITHMETIC WORD PROBLEMS

by

Mark D. LeBlanc
B.A., University of Maine, 1984
M.S., University of New Hampshire, 1987

DISSertation

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy
in
Engineering

May, 1993
This dissertation has been examined and approved.

Sylvia Weber Russell
Dissertation Director, Dr. Sylvia Weber Russell
Adjunct Associate Professor of Computer Science

Dr. R. Daniel Bergeron
Professor of Computer Science

Joan Ferrini-Mundy
Dr. Joan Ferrini-Mundy
Associate Professor of Mathematics

Filson H. Glanz
Dr. Filson H. Glanz
Professor of Electrical Engineering

Edward J. O’Brien
Dr. Edward J. O’Brien
Associate Professor of Psychology

Date
5/2/93
How concerning to me are your thoughts, O God!
How vast is the sum of them!
Were I to count them,
they would outnumber the grains of sand.

Psalm 139:17-18a
DEDICATION

To my best friend and wife, Kathleen Mary
ACKNOWLEDGEMENTS

I wish to give special thanks to my dissertation committee members, Dr. Sylvia Weber-Russell, Dr. R. Daniel Bergeron, Dr. Joan Ferrini-Mundy, Dr. Filson Glanz and Dr. Edward O’Brien. Their personal interests in interdisciplinary research provided the encouragement and flexibility I needed to commit my dissertation project to this area. Extra thanks go to Sylvia for her wisdom and patience.

My sincere thanks also go to Jim Burns for taking a rough prototype of the sentence-level parser and turning it into a tool for present and future research. Jim spent countless hours updating and generalizing the parser in order to coordinate my design of the entire model.

And last, my deepest thanks to my family for teaching Papa that research is just a hobby, no matter what the world tells me. The extra time it took to finish this work was a direct result of time spent wrestling on the living room rug. The outcome from time well spent with them will be evident long after the results of this work.
# TABLE OF CONTENTS

DEDICATION ......................................................................................................................... iv

ACKNOWLEDGEMENTS ......................................................................................................... v

LIST OF TABLES .................................................................................................................... x

LIST OF FIGURES .................................................................................................................. xiv

ABSTRACT ............................................................................................................................. xv

Chapter 1: Introduction .......................................................................................................... 1

Chapter 2: Literature Review ................................................................................................. 5

2.1. Word Problem Research: An Interdisciplinary Effort ....................................................... 5

2.2. Isolating The Sources of Word Problem Difficulty ......................................................... 6

2.2.1. Structural Format Variables ...................................................................................... 7

2.2.2. The Influence of Syntax ........................................................................................... 8

2.2.3. Readability Formulas ............................................................................................... 9

2.2.4. Classification by Semantic Structure ....................................................................... 9

2.2.5. Information-Processing Models ............................................................................... 12

2.2.5.1. The Direct Translation Influence ...................................................................... 12

2.2.5.2. The Logico-Mathematical View of Problem Solving ......................................... 13

2.2.5.3. A Linguistic Development View of Problem Solving ......................................... 15

2.2.5.4. Towards a Model of Text Integration in Word Problem Solving ......................... 16

2.2.6. Text Comprehension ................................................................................................. 17

2.2.7. The Importance of Problem Wording ....................................................................... 18

vi
4.1.1. Method .......................................................................................................... 58

4.1.1.1. Materials .................................................................................................. 58

4.1.1.2. Procedure .............................................................................................. 59

4.1.1.3. Predictor Variables ............................................................................... 60

4.1.2. Results ....................................................................................................... 64

4.1.3. Discussion .................................................................................................. 66

4.2. Experiment 2 ................................................................................................. 68

4.2.1. Method ...................................................................................................... 68

4.2.1.1. Materials and Procedure ....................................................................... 69

4.2.1.2. Predictor Variables ............................................................................... 69

4.2.2. Results ....................................................................................................... 70

4.2.3. Discussion .................................................................................................. 72

4.3. Experiment 3 ................................................................................................. 74

4.3.1. Method ...................................................................................................... 74

4.3.2. Results ....................................................................................................... 74

4.3.3. Discussion .................................................................................................. 75

4.4. General Discussion ....................................................................................... 76

Chapter 5: A Network of Problems ...................................................................... 79

5.1. Introduction ................................................................................................... 79

5.2. Task Characteristics of Problem Difficulty .................................................. 80

5.2.1. Semantic Structure and Location of the Unknown ..................................... 80

5.2.2. Action Type .............................................................................................. 82

5.2.3. Situational Characteristics ........................................................................ 83
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.3. Change 3</td>
<td>120</td>
</tr>
<tr>
<td>8.4. Change 4</td>
<td>121</td>
</tr>
<tr>
<td>8.5. Change 5</td>
<td>121</td>
</tr>
<tr>
<td>8.6. Change 6</td>
<td>122</td>
</tr>
<tr>
<td>8.7. Combine 1</td>
<td>122</td>
</tr>
<tr>
<td>8.8. Combine 2</td>
<td>123</td>
</tr>
<tr>
<td>8.9. Combine 3</td>
<td>123</td>
</tr>
<tr>
<td>8.10. Combine 4</td>
<td>124</td>
</tr>
<tr>
<td>8.11. Combine 5</td>
<td>124</td>
</tr>
<tr>
<td>8.12. Combine 6</td>
<td>125</td>
</tr>
<tr>
<td>8.13. Compare 1</td>
<td>125</td>
</tr>
<tr>
<td>8.14. Compare 2</td>
<td>126</td>
</tr>
<tr>
<td>8.15. Compare 3</td>
<td>126</td>
</tr>
<tr>
<td>8.16. Compare 4</td>
<td>127</td>
</tr>
<tr>
<td>8.17. Compare 5</td>
<td>127</td>
</tr>
<tr>
<td>8.18. Compare 6</td>
<td>128</td>
</tr>
<tr>
<td><strong>Appendix C: Additional Network of Problems</strong></td>
<td>129</td>
</tr>
<tr>
<td>9.1. Change Problems</td>
<td>129</td>
</tr>
<tr>
<td>9.1.1. Making Change 1 Problems Easier and Harder</td>
<td>129</td>
</tr>
<tr>
<td>9.1.2. Making Change 2 Problems Easier</td>
<td>129</td>
</tr>
<tr>
<td>9.1.3. Making Change 2 Problems Harder</td>
<td>131</td>
</tr>
<tr>
<td>9.2. Making Change 3 Problems Easier</td>
<td>136</td>
</tr>
<tr>
<td>9.3. Making Change 3 problems harder</td>
<td>139</td>
</tr>
<tr>
<td>Chapter/Section</td>
<td>Title</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>9.4</td>
<td>Making Change 4 Problems Easier</td>
</tr>
<tr>
<td>9.5</td>
<td>Making Change 4 Problems Harder</td>
</tr>
<tr>
<td>9.6</td>
<td>Making Change 5 Problems Easier</td>
</tr>
<tr>
<td>9.7</td>
<td>Making Change 5 Problems Harder</td>
</tr>
<tr>
<td>9.8</td>
<td>Making Change 6 Problems Easier</td>
</tr>
<tr>
<td>9.9</td>
<td>Making Change 6 Problems Harder</td>
</tr>
<tr>
<td>10</td>
<td>Combine Problems</td>
</tr>
<tr>
<td>10.1</td>
<td>Making Combine 1 and 2 Problems Easier</td>
</tr>
<tr>
<td>10.2</td>
<td>Making Combine 1 and 2 Problems Harder</td>
</tr>
<tr>
<td>10.3</td>
<td>Making Combine 3 and 4 Problems Easier</td>
</tr>
<tr>
<td>10.4</td>
<td>Making Combine 3 and 4 Problems Harder</td>
</tr>
<tr>
<td>10.5</td>
<td>Making Combine 5 and 6 Problems Easier and Harder</td>
</tr>
<tr>
<td>11</td>
<td>Compare Problems</td>
</tr>
<tr>
<td>11.1</td>
<td>Making Compare 1 Problems Easier</td>
</tr>
<tr>
<td>11.2</td>
<td>Making Compare 1 Problems Harder</td>
</tr>
<tr>
<td>11.3</td>
<td>Making Compare 2 Problems Easier</td>
</tr>
<tr>
<td>11.4</td>
<td>Making Compare 2 Problems Harder</td>
</tr>
<tr>
<td>11.5</td>
<td>Making Compare 3 and 4 Problems Easier</td>
</tr>
<tr>
<td>11.6</td>
<td>Making Compare 5 and 6 Problems Easier</td>
</tr>
<tr>
<td>11.7</td>
<td>Making Compare 5 and 6 Problems Harder</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Implementation Details</td>
</tr>
<tr>
<td>Bibliography</td>
<td></td>
</tr>
</tbody>
</table>
# LIST OF TABLES

1. Summary of EDUCE’s Processing for "Jake has" .......................................................... 28
2. Summary of EDUCE’s processing for "Jake has 5 soda cans." ......................................... 29
3. Summary of SELAH’s processing on an "of them" problem ............................................... 37
4. Summary of SELAH’s processing on an action problem .................................................. 44
5. Summary of SELAH’s processing on a non-relational problem ......................................... 50
6. A summary of SELAH’s processing on a relational "more than" problem ............................. 54
7. A summary of SELAH’s processing on a relational "fewer than" problem ........................... 56
8. Proportions of correct solutions (with blocks) .................................................................. 60
9. Proportions of correct solutions (without blocks) ............................................................ 61
10. Example of scoring for AVGMEM and AVGINF variables ............................................... 62
11. Actual scores of the predictor variables used in Experiment 1 ......................................... 64
12. Summary of the predictor variables used in Experiment 1 ............................................... 65
13. Results of regression analysis without blocks in Experiment 1 ......................................... 65
14. Results of regression analysis with blocks in Experiment 1 ............................................. 66
15. Actual scores of the predictor variables used in Experiment 2 ......................................... 71
16. Summary of the predictor variables used in Experiment 2 ............................................... 71
17. Results of regression analysis without blocks for Experiment 2 ....................................... 72
18. Results of regression analysis with blocks for Experiment 2 ........................................... 72
19. Results of Regression Analysis without Blocks in Experiment 3 ..................................... 75
20. Results of Regression Analysis with blocks in Experiment 3 ........................................... 75
<table>
<thead>
<tr>
<th>Section Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proportions of correct solutions for varying semantic structures</td>
<td>81</td>
</tr>
<tr>
<td>Summary of model's processing for traditional Change 1</td>
<td>91</td>
</tr>
<tr>
<td>Summary of model's processing for time-altered Change 1</td>
<td>92</td>
</tr>
<tr>
<td>Predicted and observed 1st grade probability of solutions for Change 1</td>
<td>93</td>
</tr>
<tr>
<td>Predicted and observed 3rd grade probability of solutions for Change 1</td>
<td>94</td>
</tr>
<tr>
<td>Change 1 (result unknown-JOIN) reworded versions</td>
<td>96</td>
</tr>
<tr>
<td>Predicted and observed 1st and 2nd grade probabilities for reworded versions of Combine 5</td>
<td>99</td>
</tr>
<tr>
<td>Predicted and observed 1st grade probabilities for reworded versions of Combine 5</td>
<td>101</td>
</tr>
<tr>
<td>Predicted and observed probabilities of solution for Combine 5 and 6</td>
<td>103</td>
</tr>
<tr>
<td>Combine 5 and 6 (subset unknown--SEPARATE-FROM) reworded versions</td>
<td>104</td>
</tr>
<tr>
<td>Comparison of Average Solution Results Across Grades for Compare 3 and 4 Problems</td>
<td>106</td>
</tr>
<tr>
<td>Comparison of predicted and observed Compare 3 probabilities and the equalize version</td>
<td>107</td>
</tr>
<tr>
<td>Compare 3 and 4 (compared quantity unknown) reworded versions</td>
<td>108</td>
</tr>
<tr>
<td>Subset of problems from the network for “addition with the superset unknown”</td>
<td>110</td>
</tr>
<tr>
<td>Summary of model's processing for traditional Change 2</td>
<td>132</td>
</tr>
<tr>
<td>Summary of model's processing for time-altered Change 2</td>
<td>133</td>
</tr>
<tr>
<td>Predicted and observed probability of solutions for time-altered Change 2</td>
<td>134</td>
</tr>
<tr>
<td>Change 2 (Result Unknown-SEPARATE-FROM) reworded versions</td>
<td>135</td>
</tr>
<tr>
<td>Comparison of Riley and Carpenter versions of Change 3 and predictions for second grade</td>
<td>138</td>
</tr>
</tbody>
</table>

xiii
### LIST OF FIGURES

1. The fundamental transitions between representations facing the word problem solver .............................. 24
2. A simplified version of EDUCE’s lexical entry for the word “altogether” .................................................. 32
3. SELAH’s initial distinctions between conceptual representation components produced by EDUCE ................................................................. 39
4. A sample arithmetic word problem ........................................................................................................ 58
5. A problem involving a potential change from JOIN to SEPARATE_FROM ................................................. 67
6. Chronologically presented time sequence ............................................................................................... 84
7. Non-chronological time sequence ........................................................................................................ 84
8. Traditional Change 1 ............................................................................................................................... 86
9. Change 1 with consistent narrative focus ............................................................................................... 87
10. Change 1 with related co-actor ............................................................................................................ 87
11. Removing (time language) from Change 1 ........................................................................................... 89
12. Altering the time sequence of Change 1 .............................................................................................. 90
13. Equations predicting probability of solution for first and third grade .................................................. 93
14. Traditional Combine 5 ........................................................................................................................ 96
15. Including of them and the rest to make Combine 5 easier .................................................................. 97
16. Including set reference language and removing altogether in Combine 5 .......................................... 98
17. Traditional Combine 6 ........................................................................................................................ 101
18. Traditional Compare 3 ........................................................................................................................ 103
19. Traditional Compare 4 ........................................................................................................................ 104
ABSTRACT

A COMPUTER MODEL OF THE ROLE OF TEXT INTEGRATION
IN THE SOLUTION OF ARITHMETIC WORD PROBLEMS

by

Mark D. LeBlanc

University of New Hampshire, May, 1993

Understanding arithmetic word problems involves a complex interaction of text comprehension and mathematical processes. This work presents a computer model of the hypothesized processes that are required of a young student solving arithmetic word problems, including the processes of sentence-level reading and text integration. Unlike previous computer simulations of word problem solving, which neglect the early stages of text processing, this model forces a detailed consideration of the linguistic process, which is being increasingly recognized as a primary source of difficulty. Three experiments were conducted to isolate critical text comprehension processes. Children's probability of solution was analyzed in regression analyses as a function of the model's text comprehension processes. A variable measuring the combined effects of the load on working memory and text integration inferences accounted for a significant amount of variance across four grade levels (K-3). The results suggest new process-oriented measures of determining why a particular word problem may be difficult, especially for young students. An implication for education is the potential for a difficulty-differentiated network of problems that includes a multiple number of rewordings for each "traditional" problem wording as an aid for classroom assessment, curriculum development and future computer-based learning environments.
CHAPTER 1

Introduction

This thesis presents a detailed computer model of arithmetic word problem solving to investigate the linguistic and conceptual difficulties young children experience in learning mathematics. The outcome is new, more fine-grained measures of problem difficulty and sequencing as an aid to teachers in assessing and guiding their students, as a contribution to curriculum development, and as a potential component in the system design of computer learning environments.

Relating natural language to mathematical language is an important component of elementary mathematics education. The mediating role of natural language in expressing everyday problem situations is increasingly being recognized as a potential focus for both research on children's mathematical problem solving and new classroom approaches to teaching "mathematical thinking." The National Council of Teachers of Mathematics (NCTM) Curriculum (1989) and Professional (1991) Standards emphasize the pedagogical importance of representing and reading mathematics.

Traditionally, the verbal aspect of mathematics at the elementary level has been incorporated into curricula as "word problems," which have often served as a source of children's continuing frustration with mathematics. The recent emphasis on how children could or do make connections between natural language and mathematical concepts provides new motivations for a consideration of why certain linguistic forms of mathematical problem situations are difficult. Carpenter, Fennema, Peterson and Carey (1988) note that most teachers' knowledge of children's thinking is not organized into a coherent network which distinguishes different types of problem difficulty. An underlying assumption in this research is that teachers who present problems would benefit from an awareness of how the phrasing of those problems can affect the relative difficulty of those problems. This approach to identifying types of problem
difficulty goes beyond the traditional problem classification by semantic structure, considered by some researchers (e.g., Cobb, Yackel & Wood, 1988) to be an incomplete form of pedagogical knowledge.

There is as yet no detailed hypothesis of how a student might proceed from linguistic (mis)understanding to mathematical (mis)understanding. A precise model of this complex process, i.e., a computer model, starting with the initial presentation of the problem would be consistent with Goldin's (1992) call for a comprehensive model of mathematical problem understanding. In particular, a model which focuses on early stages of processing would force a consideration of critical features of the problem solving process that are often overlooked and could suggest specific processing variables relating to problem difficulty. A correlation of such variables with problem difficulty could in turn be used to establish a hypothesized network of types of linguistic variations representing different dimensions and levels of difficulty. In addition, the future use of such a "reading" model in a computer learning environment oriented toward conceptual learning and exploration would facilitate the desirable goal of allowing children to formulate verbalizations of their own problems. Children could see their problems processed in multiple modes of representation, and in general to see the lexical mode of a problem representation dynamically linked to other modes of representation, as advocated by Kaput (1992).

The information processing and learning assumptions in this research reflect the ideas 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 (i) highlight certain mathematical relationships, e.g., direct references to previous sets facilitates the process of text integration, and/or (ii) lead to situational interpretations of events or quantities which are easier to retain in memory. In order to encourage progress without making jumps for which the student is not ready, problem sequencing should depend on verbal presentation of the problem as well as mathematical structure. A more fine-grained network of problem types in terms of dimensions and levels of difficulty reflecting such problem rewordings can assist in both curriculum design and assessment of students by
teachers.

The domain of this research is restricted to verbal problems that describe a variety of situations involving the arithmetic actions of changing, combining, comparing, and equalizing, expressed as one-step addition and subtraction word problems typical of those presented to children of age six to eight. The following steps outline the research to be described:

1. **Construct a computer program representing a detailed computer model** of the hypothesized tacit reading and text integration processes which lead from a linguistic word problem statement to its solution in terms of correct mathematical relationships. The system design of the computer model (see Chapter 3) involves two interacting components which disambiguate the English problem statement in terms of conceptual structures.

2. **Isolate potential text comprehension tasks** in the model that are predictive of problem difficulty such as the number of required inferences and the load on working memory (see Chapter 4). Conduct studies of the model to test hypotheses concerning the correlation between these tasks and children's previously established probabilities of solution.

3. **Conduct studies with regression analyses based on the model's performance in (2)** to predict the effects of slight rewordings on the probability of solution. Extend these results to create a difficulty-differentiated network of problems that includes a multiple number of rewordings for each traditional problem wording (see Chapter 5). The relative difficulties assigned to the multiple rewordings will be determined by the predictions of the model.

The fundamental hypothesis of the computer model presented here is that children's ability to follow-up on explicit set references (or infer such references) is a crucial step towards recognizing the conditions that make an arithmetic operation appropriate for a given situation. The empirical results present new measures of text comprehension which determine why a particular word problem may be difficult, especially for young readers. In this sense, the computer model proposes a new fine-grained
classification of problems which vary according to both logical-mathematical problem solving processes and the often overlooked but critical processes involved in text comprehension.
CHAPTER 2

Literature Review

Arithmetic word problems\(^1\) are a classic example of an area of mathematics with direct practical application. Real-world problems do not come packaged in equations ready to be solved but rather as verbal and pictorial representations that must be understood, abstracted and mapped onto symbolic representations that can be solved. It is primarily for this reason that arithmetic word problems play such a vital role in early mathematics education.

Following an introduction on the interdisciplinary nature of the research on arithmetic word problems, this chapter presents the evolution of methods which attempted to predict the probability of solution for particular types of problems. Of specific interest is the move away from static methods (e.g., syntactic complexity) to process-oriented measures (logical-mathematical and linguistic processing). In closing, the chapter highlights the need for a detailed look at the role of text integration in word problem solutions, specifically in problem wordings which do not contain significant actions.

2.1. Word Problem Research: An Interdisciplinary Effort

On the basis of interdisciplinary research of the previous decade, a new focus is emerging in the attempt to address the complexities of children’s word problem solving difficulties. Researchers in cognitive psychology, artificial intelligence and mathematics education are unifying the results of interview and observational studies, cognitive simulation models, and eye movement and recall experiments in order to focus attention on the cognitive dimensions involved in word problem solutions.

\(^1\) An arithmetic word problem can be defined as a small written paragraph of text describing the mathematical relationships between a certain number of quantities where one or more of the quantities is unknown and its value is requested.
The idea of focusing on various cognitive dimensions of word problem solution rather than simply on problem classification receives support from interdisciplinary research concerned with, e.g., (i) the recognition of semantic relationships (Booth, 1989; DeCorte et al., 1990; Herceovics, 1989; Kaput, 1987; and many others), (ii) the use of problem-solving strategies (Thompson, 1988; Greeno, 1987) to encourage children to find mathematical meaning by constructing their own solutions (Steffe, Cobb & von Glasersfeld, 1988; Cobb, Yackel & Wood, 1988) as opposed, for example, to superficial "clue-word" approaches, (iii) children's expression of how they solve problems as opposed to a sole concern for producing the correct answer (Lampert, 1988), (iv) the role of integrated propositions in memory (Trabasso & Sperry, 1985, and others) and (v) the importance of short-term memory as a bottleneck in the comprehension process (Fletcher, 1986; Kintsch and Greeno, 1985). However, linguistic factors have only recently been recognized as a level of representation at which misunderstandings can occur (Cummins, Kintsch, Reusser & Weimer, 1988; DeCorte, Verschaffel, & DeWitt, 1985; Stern, 1993 and others). Neglect of the initial linguistic representation of a problem is reflected in the fact that existing computer simulations of word problem solution skip the stage of the verbal problem statement and start directly with artificial propositions.

The common denominator in this interdisciplinary research is the relatively unexplored processes of qualitative reasoning in children's mathematics word problem solving. This research is contributing to a better and more fine-grained understanding of the qualitative reasoning processes involved in children's solution processes. Its instructional importance lies in the crucial role of understanding why slight variations in traditional "textbook" problem wordings can improve young children's problem solving success.

2.2. Isolating The Sources of Word Problem Difficulty

Word problems are regarded as an early indicator of problem solving performance (NCTM, 1989). Starting as early as first grade, curriculum materials place arithmetic word problems at the end of sections which focus on specific arithmetic skills. For example, open-sentence computations like, 5 + 3 = ?, are
followed by a section of word problems such as:

Jake had 5 soda cans. Mary gave him 3 more soda cans. How many soda cans does Jake have now?

or the more complicated but arithmetically equivalent problem:

Jake has some soda cans. He gave Mary 5 soda cans. Now Jake has 3 soda cans. How many soda cans did Jake have in the beginning?

The above two problems are indeed arithmetically equivalent, that is, they both potentially involve $5 + 3 = ?$ as a computational solution. However, most researchers, teachers, and students(!) notice that the second problem is considerably more difficult than the first (Carpenter, Fennema et al., 1988). But exactly what is it that generates this difference in difficulty? The focus of this literature review is to highlight research studies which address the following question:

Why are some word problems more difficult to solve than others?

The studies to be reviewed measure problem difficulty with (1) structural format variables, (2) syntactic structure, (3) readability formulas, and (4) semantic complexity. In addition, studies use information-processing models to relate problem difficulty to the degrees of expertise necessary to solve various problems, including (5) mathematical, (6) linguistic and (7) situational expertise. Finally, new theories in (8) text comprehension complement a recent emphasis on (9) the importance of problem wording, especially when the choice of words facilitates text integration.

2.2.1. Structural Format Variables

Early attempts to isolate the sources of word problem difficulty centered around the use of structural format variables as predictors of student-word-problem-solving performance (Searle, Lorton & Suppes, 1974). A sample of structural variables includes: the complexity of the vocabulary, the presence of extraneous information, the order of numerical information, and the presence of a diagram, to name but a few. The motivation behind the studies was threefold: (i) to isolate structural variables; (ii) to use those
variables to structure the curriculum (i.e., put the easiest problems first); and (iii) to predict student performance. Some attempts were also made to teach students to identify and modify a specific structural variable in order to reduce the difficulty of a particular problem (Cohen and Stover, 1981).

The central contribution of the structural variable approach for explaining word problem difficulty is the notion that many different variables influence the difficulty of a particular problem. While the variables are static in nature, that is, they focus on the problem structure rather than on the problem solving processes, they provided an early glimpse of the detail which would be required in order to fully model the complexity of word problem solving.

Missing from the structural variables studies, however, are variables which indicate the role of text comprehension during word problem solution. The limited set of “linguistic” variables only include descriptions of the problem statement such as: the length of the longest word, the number of words in the longest sentence, and the number of sentences in a problem. As shown by Jerman and Sanford (1974), none of these variables accounted for the observed variance in proportion of correct solution across grade levels.

2.2.2. The Influence of Syntax

In addition to structural variables which stress the role of unfamiliar vocabulary and the length of the longest sentence, a number of studies investigated the role of sentence syntax on word problem difficulty and how syntax can be used to obtain information about a problem solver’s competencies (Golddin & McClintock, 1984). The results from such studies are mixed. While some studies were able to show that problems consisting of difficult syntactic constructions were harder than simple sentences (Wheeler and McNutt, 1983), other studies showed no effect from syntactic changes (Muth, 1982). Unlike the relatively differentiated sources of difficulty in the structural format variable studies, the syntactic variable is a coarse-grained measure of solution probability, distinguishing only between simple, complex and compound-complex sentences. Even in studies which do show a positive correlation
between increasing syntactic complexity and problem difficulty, the conclusion is narrow: problems with compound-complex sentences are more difficult than problems with only simple sentences.

2.2.3. Readability Formulas

Readability formulas are a widely used and often criticized (e.g., Fitzgerald and Cullinan, 1984) method of gauging the difficulty of passages of connected prose. Typically, these formulas are a function of average sentence length and difficulty of vocabulary. While two early studies (Linville, 1976; Thompson, 1967) claimed that making mathematical word problems easier to read raises achievement level, Paul, Nibblelink and Hoover (1986) have shown that this is not entirely true. By drawing on some of the structural format variables, Paul et al. were able to show that the Linville (1976) and Thompson (1967) studies did not isolate readability as the independent variable. Paul et al.'s work, in particular, showed no effect of readability on students' ability to solve word problems. This conclusion is not surprising since readability formulas were never intended to be used for specialized materials such as mathematical word problems. Text comprehension theories typically caution that specialized texts such as mathematical word problems require specialized reading strategies (Kintsch and Greeno, 1985; Reusser, 1989).

2.2.4. Classification by Semantic Structure

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. 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. More specifically, classifying problems by semantic structure is a function of the following two factors: (i) the presence (or absence) of action verbs and (ii) children's solution strategies. Initially proposed by Heller and Greeno (1978) and generally consistent with other classification schemes (Carpenter and Moser, 1982; Mayer, 1981; Nesher and Katriel, 1977), addition and subtraction word problems
are classified into one of four types of semantic structure: Change, Combine, Compare, or Equalize.²

**Change** problems involve changes in location or possession, for example, a transfer of objects from one person to another, and describe situations that occur over time:

1. Jacob had 5 soda cans.³ Then Kathy gave him 3 soda cans. How many soda cans does Jacob have now? (Change)

**Combine** problems involve static descriptions of the numerosity of two disjoint (sub)sets and the union of those two sets, for example, problem (2). **Combine** problems do not contain action verbs.

2. Jacob found 3 pepsi cans. Kathy found 7 coke cans. How many soda cans did they find altogether? (Combine)

**Compare** problems involve the static comparison of the numerosity of two disjoint subsets, for example, problem (3). Like **Combine** problems, **Compare** problems do not contain action verbs.

3. Jacob has 5 soda cans. Kathy has 3 more soda cans than Jacob. How many soda cans does Kathy have? (Compare)

**Equalize** problems are a hybrid of **Change** and **Compare** problems (Carpenter and 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). **Equalize** problems contain action verbs and describe situations that occur over time.

4. Jacob found 4 cans. If he finds 3 more cans, he will have the same number of cans as Kathy. How many cans does Kathy have? (Equalize)

In addition to the presence or absence of action verbs, the classification of semantic structure is also based on longitudinal studies of children’s solution strategies. Developmental differences in children’s

---

² For more elaborate descriptions of the research which led to this classification see Carpenter and Moser (1982). See Appendix A for a complete set of the various types of problems within each semantic structure.

³ This research is part of a larger effort to develop an intelligent learning environment. The proposed lesson theme focuses on litter collection and recycling. The real-world forum of the problems can increase student interest in the problem-solving process (Davis-Dorsey, Ross and Morrison, 1991; Ross and Anand, 1987) and assist students with internally recognizing critical relationships between sets (Mayer, 1984).
solution strategies concern (i) direct modeling actions, (ii) counting strategies and (iii) number facts. Direct modeling actions, such as moving blocks around on a table, involve, in general, an increase, decrease, combination or comparison of physical sets of objects. The five direct modeling actions include the addition strategy of (1) “Counting All” (e.g., represent both sets using physical objects or fingers and then count the union of the two sets) and the subtraction strategies of (2) “Separating From,” (3) “Separating To,” (4) “Adding On,” and (5) “Matching” (Carpenter, 1985). In addition, each direct modeling strategy has an associated mental counting strategy. Counting strategies do not involve physical aids and are therefore a more efficient and less mechanical application of a solution technique. For example, the direct modeling strategy of “Counting All” has an associated counting strategy, “Counting On From First.” To arrive at the sum of 5 apples and 3 apples, a child may decide to start the counting at the first number in the problem (5 ... 6, 7, 8) rather than physically making a group of 8 apples and then “Counting All” of the apples. As pointed out by Carpenter and Moser (1983), Fuson (1982) and others, children progress from concrete direct modeling strategies to more abstract and efficient counting strategies to the eventual use of number facts.

Because semantic structure reflects children’s problem solving strategies (Riley, Greeno and Heller, 1983; Carpenter and Moser, 1982), semantic structure is a move beyond earlier attempts to base problem difficulty solely on the surface structure of problems, such as structural format variables or syntax. 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 (Carpenter and Moser, 1984; Cummins et al., 1988; DeCorte & Verschaffel, 1987; DeCorte, Verschaffel & Pauwels, 1990; Rathmell, 1986; Riley et al., 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 in grades K-3. On the average, Change 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 identity of the unknown quantity (Hiebert, 1982).

2.2.5. Information-Processing Models

Information-processing models of arithmetic word problem solving (Briars and Larkin, 1984; Cummins et al., 1988; Kintsch and Greeno, 1985; Okamoto, 1992; Reusser, 1989; Riley et al., 1983) are the culmination of a large body of research emphasizing the relation between a problem's semantic structure, children's problem solving strategies and children's potential knowledge representations. Information-processing models are computer programs which attempt to simulate specific types of problem solving actions required for word problem solutions. By implementing a model of problem solving in a computer program, one is confronted with many of the tacit processes which are typically ignored, in line with Newell and Simon's (1972) emphasis on the integrated activities that constitute problem solving. The following sections introduce a number of computer simulations which solve word problems with a focus on the how each model attempts to isolate the sources of problem difficulty.

2.2.5.1. The Direct Translation Influence

A naive conception of problem solving that is often reflected in the mathematical curriculum is that word problems can be represented by syntactic translations of words into mathematical symbols (Burton, 1988; Carpenter, 1985; Clement, 1982). Early theories explaining the behavior of subjects solving word problems, at least to a reasonable first approximation, were based on the direct translation systems of Garfinkel (1960) and Bobrow (1968). Garfinkel developed and hand-simulated a program for translating one-variable algebra word problems into equations. Bobrow constructed a running computer program called STUDENT which solves a considerable range of algebra word problems, allowing for more than one variable and simultaneous equations. Although STUDENT did not intend to simulate human problem solving, Paige and Simon (1966) conducted an analysis which shows that some processes in the pro-
gram do, in fact, parallel human processes in important respects. Two examples are the way both STUDENT and humans assign variables to entities with no given value and the way a phrase such as "twice a number" is converted to (2 x number). Hinsley, Hayes and Simon (1977) show that students who do not recognize a category for a problem will employ as a fall-back strategy a general solution procedure which closely resembles the direct translation strategy employed by STUDENT. Yet more importantly, Paige and Simon (1966) point out that direct translation strategies are inadequate as a processing scheme alone. Semantic and auxiliary information is also needed. These findings accurately reflect Polya’s (1957) point of view concerning the inadequacy of pure syntactic mappings, as expressed in his English-to-French analogy:

An English sentence is relatively easy to translate to French if it can be translated word for word. But there are English idioms which cannot be translated into French word for word. If our sentence contains such idioms, the translation becomes more difficult; we have to pay less attention to the separate words, and more attention to the whole meaning before translating the sentence, we may have to rearrange it.

2.2.5.2. The Logico-Mathematical View of Problem Solving

The logico-mathematical view of problem solving contends that the locus of children’s improvement in solving word problems is the acquisition of knowledge concerning logical set relations, in particular, part-whole relations. An early logico-mathematical view of arithmetic word problem solving is the computational model of Riley, Greeno and Heller (1983). This model simulates a developmental progression of the types of problem schemata (knowledge representations) that are needed in order to solve Change problems which vary in solution difficulty. The model relies on a combination of problem schemata, action schemata (manipulative actions) and strategic schemata (plans). Problem schemata serve as representations of the problem situation and actions and plans determine a solution procedure. By func-

---

4 The word *model* is often used in two conflicting ways. On the one hand, it means a general problem solving theory, while on the other hand it means a computer simulation of a specific part of the general theory. Because of the emphasis on computer simulations in this section and throughout this work, I will use the latter definition. From this point on, the word "model" refers to a specific computer implementation unless noted otherwise.
tioning in one of three possible modes, (1) novice, (2) intermediate or (3) expert, the model uses varying degrees of problem schemata to represent varying levels of logico-mathematical competence:

[Novice simulations] understand quantitative relations by means of a simple schema that limits its representations of change problems to the external display of blocks. [Intermediate simulations] have a change schema for maintaining an internal representation of increases and decreases in the sets of blocks it manipulates. [Expert simulations] ... have an understanding of part-whole relations (Riley et al., 1983, p.176).

For example, at the novice level, the model has access to a schema for representing a set of objects but the novice level has neither a schema to represent the initial, change and resulting sets in a transfer nor the knowledge of part-whole relations. The three levels of knowledge in the model predict patterns of correct/wrong answers for each of the three basic semantic categories, although the model’s predicted patterns are not totally in line with other results (e.g., DeCorte & Verschaffel, 1991; Del Campo and Clements, 1987).

The critical theoretical feature of this model and the more advanced models which also solve Combine and Compare problems (Briars and Lurkin, 1984; Kintsch and Greeno, 1985; Riley and Greeno, 1988) is the emphasis that problem competence is directly related to the presence of part-whole knowledge. In other words, if a child can not solve a problem that the model solves with part-whole knowledge, these models hypothesize that the child does not yet understand part-whole relations. This claim has recently met stiff opposition, due in a large part to the restricted set of problem wordings solved by the model. Cummins (1991), Hudson (1983) and others have shown that children as young as 4-years old exhibit a sophisticated grasp of part-whole relations on problems worded in a certain way and yet can not solve the traditional wordings solved by the models. Clearly, these children can use their working knowledge of part-whole relations if they can map relational expressions in the problem statement onto that knowledge. As pointed out by Cummins (1991), “the (part-whole) knowledge is there, but it simply is not accessed when problems are worded in certain ways” (p.267).
In short, the models of Briars and Larkin (1984), Kintsch and Greeno (1985), Riley et al. (1983) and Riley and Greeno (1988) offered new hypotheses about the relationships between problem representations and solution procedures at varying levels of competence. A critical shortcoming of these models, however, is the insensitivity to small changes in problem wording. A linguistic development view begins to address this issue.

2.2.5.3. A Linguistic Development View of Problem Solving

Since problem representations play such a vital role in the models' solutions, the linguistic processes which generate problem representations are now recognized as vital components in the overall problem-solving process. More specifically, the model of Cummins et al. (1988) isolates three types of knowledge needed in word problem solutions: (i) logico-mathematical knowledge (e.g., part-whole knowledge), (ii) situation knowledge (e.g., a transfer(change) schema) and (iii) lexical knowledge (e.g., the meaning of individual words). The model, an extension of earlier implementations (Dellarosa, 1985; Fletcher, 1985; Dellarosa, 1986) removes one of the three different types of knowledge at a time and solves the same set of problems as a group of children. The experimental procedure was to require the model to attempt the problems under conditions of impaired knowledge and compare its performance with that of children. In the Cummins et al. (1988) results, students' incorrect solutions most closely matched the model's solutions when lexical information was erroneous. For example, the model gives the same incorrect answers as many children when the word "some" is treated as an adjective in the model's lexicon rather than the correct mathematical-sense of a quantifier. That is, the model interprets "some bikes" like "red bikes." Similar matches between children's results and the model's simulation were found when the model interpreted the word "altogether" like the word "each" or the phrase "have more than" as simply "have."

5 A contribution of the Kintsch and Greeno (1985) model that deserves mention is their suggestion that the amount of processing required for problem solution may be related to problem difficulty. As described more fully in Chapters 3, 4 and 5, the current model makes full use of this suggestion.
The linguistic developmental view hypothesizes that children's misunderstanding of certain "mathematical" words is a major source of difficulty with word problems. Contrary to, but not excluding the logico-mathematical view, the linguistic view shifts attention away from intermediate or final stages of the problem solving process and focuses attention on the beginning processes of understanding the text (cf. Reusser, 1989).

2.2.5.4. Towards a Model of Text Integration in Word Problem Solving

The computer simulations just summarized provide a set of hypotheses concerning the mathematical and linguistic knowledge involved in word problem solutions. Using the general theory of text comprehension developed by van Dijk and Kintsch (1983), the Kintsch and Greeno (1985) model was the first to study the interaction between comprehension and problem solving. Cummins et al. (1988) focused on the importance of understanding mathematical terms. What these models lack, however, is a focus on the processes involved in a transition from natural language to a representation of the essentials of a situation which can then, in turn, be mapped onto more abstract mathematical knowledge. In short, the limitation of previous systems

lies in the almost perfect match and the *a priori* mapping between the propositional structure of the textbase and the set structure of the mathematical problem model containing, as its latent property, the solution equation (Reusser, 1987, p. 7).

Reusser's (1989) *Situation Problem Solver* (SPS) is the first attempt to model the progressive and incremental process of transformation from text to situation to equation. Unlike the previous models which rely on an unreliable key word approach (cf. NCTM, 1989; Nesher and Teubal, 1975), SPS simulates word problem solutions as a process of reconstructing a situation with an abstract mathematical point of view. Because the model relies heavily on an understanding of the situation described in the text, an elaborate text-processing component is required to distinguish the representation of the situation from the representation of the text, as advocated by Kintsch (1986). For example, SPS must recognize the protagonist of the story, natural vs. non-natural time order of events, irrelevant information and the
goal of the problem if a question is not given. In addition, SPS must separate the action part of the story from its motivational setting and its context. An "action-theoretic" approach enables SPS to arrive at a mathematical operation given a situational representation. In other words,

concrete actions, expressed (in word problems) as action verbs, are seen as bearing [containing inherently] more or less abstract mathematical operations, as pointing to specific and abstract relational ideas which can be, finally, expressed by the set of mathematical operation schemata (addition, subtraction, multiplication, division) (Reusser, 1989, p.13).

SPS stresses that from a problem solving (and instructional) point of view, arriving at a representation of the situation in a problem should not be a superfluous, but a necessary process.

Because SPS currently solves only the action-worded Change problems, the role of situational representations in statically-worded problems (e.g., Combine and Compare) is not entirely clear (cf. Stern & Lehrndorfer, 1992). If indeed a sense of mathematical operation emerges from actions in situations, a number of questions remain: "Do children attempt to interpret action-void (static) word problems from a situational perspective?" and if so, "How do children arrive at (hypothetical) situations given a text that only describes the static relationships between quantities?"

2.2.6. Text Comprehension

Based on recent research in text comprehension, an essential ingredient in a model of word problem solving is a capability to establish the mathematical connection (i.e., integrate) two sets of objects by either implicit or direct reference. One of the focal points of research in discourse understanding has been the development of models for the representation of text in memory (e.g., van Dijk and Kintsch, 1983; Rumelhart, 1975) and recent extensions to these models have emphasized the role of integrated propositions in memory (O’Brien, 1987a; Trabasso and Sperry, 1985; van den Broek, 1988 and others) and the importance of short-term memory as a bottleneck in the comprehension process (Fletcher, 1986

---

5 The set of problems solved by SPS far exceeds the traditional set of Change problems and the model represents a much-needed focus on the critical processes of text comprehension. The main point here is not to criticize SPS, but rather to focus attention on the role of situations in statically worded problems.
and others):

Within [an] interconnected representation, important propositions are assumed to differ from less important propositions by the number of interconnections or retrieval routes, rather than by [the traditional measure of importance associated with position or] height in the network (O'Brien, 1987b, p. 420).

In short, the empirical evidence from experiments using importance ratings and the time to retrieve a concept from memory are best explained by a model of text representation where important propositions are highly connected.

In the context of arithmetic word problems, a critical process during reading is the ability to establish mathematical connections between sentences. Because word problem texts are in general very brief and often do not contain action language that describes a specific situation, an ability to follow-up on explicit references to sets of objects (or infer such references) is crucial for understanding the relationships between sets and how that relationship can imply an action-oriented interpretation. One hypothesis of the computer model introduced in Chapter 3 is that children's ability to follow-up on explicit set references (or infer such references) is a crucial step towards recognizing the conditions that make an arithmetic operation appropriate for a given situation. The choice of problem wording directly affects whether the relationships between sets is explicit or implicit.

2.2.7. The Importance of Problem Wording

There is a considerable body of evidence which points to the fact that minor linguistic modifications greatly affect children's ability to solve a problem (Carpenter, Hiebert, & Moser, 1981; Davis-Dorsey et al., 1991; DeCorte, Verschaffel & DeWinn, 1985; Hudson, 1983; Nescher & Teubal, 1975). In many cases, a particular problem and its reworded version are of the same semantic structure and the unknown is located in the same sentence, yet results show higher levels of performance on reworded problems.
For example, DeCorte, Verschaffel and DeWinn (1985) had children solve the following two problems:

Tom and Ann have 9 nuts altogether.  Tom has 3 nuts. How many nuts does Ann have?

Tom and Ann have 9 nuts altogether. 3 of these nuts belong to Tom. The rest belong to Ann. How many nuts does Ann have?

The problem on the left is a traditional wording found in most classifications of Combine problems. It is a sparsely worded text and includes what Kintsch and Greeno (1985) refer to as "textual presuppositions," for example, that Tom’s 3 nuts are three of the nuts that Tom and Ann jointly possess and not 3 unrelated ones. The problem on the right is reworded to include explicit references between the sets, e.g., "3 of these" and "the rest." DeCorte et al.’s (1985) results show that first and second graders’ solution success significantly improved on the reworded versions, from 43% on the traditional wording to 57% on the reworded version for first graders and from 71% to 83% for second graders.

Of particular interest is that none of the previous computer simulations are sensitive to changes in wording, such as adding the phrases "of them" and "the rest." All of the models begin with encoded representations of the sentences in the word problem or depend on the presence of keywords and thus do not simulate the initial problem solving processes of reading and conceptualization. It is clear from the empirical evidence on rewording studies as well as the detailed results from eye movement experiments (DeCorte et al., 1990; Hegarty, Mayer & Green, 1992; Lewis & Mayer, 1987; Verschaffel, DeCorte & Pauwels, 1992) that additional, process-oriented measures of problem difficulty are needed.

2.3. Returning to the Process of Reading

Attempts to isolate the sources of difficulty in word problems have come "full circle." Early studies with structural variables recognized the unique challenges in word problem solutions, yet largely ignored the linguistic component of the word problem. Later research with syntactic variables and readability formulas attempted to shift the focus to the verbal construction of the problem, but did not account for the relationships and actions specific to comprehending word problems. Recent attention to children’s
problem solving strategies has led to a characterization which emphasizes the semantic relationships in problems and computer simulations of arithmetic word problem solving focus on the types of knowledge representations required for problems of varying semantic structures. In these models, problem difficulty is considered as a function of both logico-mathematical and lexical knowledge, where a model's interpretation of the problem statement directly affects the construction of a situation whose actions provide access to mathematical operations. Since situations are not always explicit in the text (e.g., the problem contains no significant actions), it is now evident that the processes of text integration, sensitive to slight changes in problem wording, are a vital component in the overall problem-solving process.
CHAPTER 3

A Computer Model of "Mathematical Reading"

3.1. Towards A Model of "Mathematical Reading"

Mathematical representations such as equations can account for or unify a large class of semantic structures of arithmetic word problems, each of which can in turn be expressed linguistically in many different ways. An important task from an educational perspective is to help the elementary-level student to "see" linguistic descriptions in quantitative or mathematical terms, e.g., through progressive abstraction from a particular linguistic problem statement to an equation or other solution. "Seeing" linguistic descriptions in mathematical terms can be thought of as a translation from the symbol system of natural language to another in which conceptual and mathematical relationships are correctly and precisely represented. As Janvier (1987) points out in characterizing mathematics as the "science of significant structure," much of the work of mathematics is to determine what structure is preserved in the transition from one representation to another.

Critical word problem solving tasks involve reading individual sentences and establishing qualitative connections between those sentences, which in turn are the basis for conceptualizing the quantitative relationships embedded in natural language. From the perspective of determining problem difficulty, a young reader's ability to arrive at qualitative relationships becomes a function of the language which is used to support the "mathematical connections." Recent theories have categorized problems according to their semantic structure. Lacking in the theory of semantic structure is an account of how the student's ability to arrive at mathematical connections of the problem might depend on identifiable ways in which natural language expresses or facilitates such connections.
The computer model introduced in this chapter describes how the role of text integration is being explored in a detailed simulation of word problem solving. The emphasis on reading and text integration is specific to the strategies required in reading word problems, yet goes well beyond the misleading "key-word" approaches which attempt to associate certain words and phrases with arithmetic operators. The aim of this computer model is not to translate directly from the problem statement into mathematical notation, but rather to make precise the tacit steps and processes which mediate a translation. Starting with an arithmetic word problem in English, the model "reads" and represents each word and sentence in the problem. By performing text integration across sentences via explicit and implied set references, the model builds a representation of the sets and relationships in the verbal text. Arithmetic actions are selected on the basis of the explicit or implied relationship between integrated sets, and final solution processes simulate the tasks of arriving at and carrying out an appropriate counting strategy. Reading comprehension thus overlaps with mathematical comprehension.

3.2. Starting at the Beginning

The computer program comprehends arithmetic word problems starting from the natural language statement of the problem. Researchers in mathematics education and cognitive science 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 et al., 1988; DeCorte, Verschaffel et al., 1985). That is, many difficulties occur during the comprehension of and translation from the natural language statement of the problem. By designing the computer program to parse the natural language statement in a word by word fashion, the model addresses a problem from the form of problem representation which is actually presented to the student. This design decision goes beyond lacking on linguistic comprehension prior to mathematical considerations. The assumption is not that language and mathematics are separate components, but rather that they are intertwined, or that the mathematics is "embedded" in the natural language statement. Consistent with Goldin's (1992) advo-
cacy of a global, complete model of the problem solving process, modeling the process from the begin-
ning guards against missing the roles which various representations play early in the problem solving pro-
cess.

Modeling the process of reading word problems confronts the relationship between natural language
and mathematical language. We know intuitively what natural language is. French is considered to be a
language different from that of German, Russian, or Swahili. In the field of natural language processing,
various theories have been devised to account for the translation from one into the other in terms of com-
puter representations. For example, Schank’s (1975) conceptual dependency representation is posited as a
“language free” level of thought through which equivalent paraphrases of sentences in the same or other
languages pass in the process of translation.

We also speak of the “language of mathematics,” and in fact, mathematics is sometimes viewed as
foreign language (Ervynck, 1992). Translating into this language of mathematics, or into some representa-
tion which indicates word problem understanding is often an insurmountable task for the young student
who tries to solve arithmetic word problems. Such difficulties also appear to remain at higher educational
levels. For example, Burton (1988) points out that college-level students often proceed in a haphazard,
bottom-up way to link words in the problems statement to variables of a possible equation, rather than
looking for semantic structure, e.g., in terms of “equivalence” in the language of the word problem.

The task of the student, as shown in Figure 1, is to arrive at a correct mathematical representation
having started from a situation expressed in natural language. As such it seems reasonable to represent the
process of “translation” from natural language to mathematical language as passing through a “concep-
tual level” which is free of the particular symbolism of either one of the languages, as Schank (1975) has
theorized for natural language translation and understanding. A word problem solver needs to dispense
with the particular details of the natural language in the process of arriving at increasingly abstract
representations of the mathematical relationships in the situation. A basic step in designing a process
model of this type of extraction/abstraction is to include components which (i) read and understand the
(sometimes mathematical) meaning and role of individual words, (ii) arrive at an unambiguous representa-
tion for each sentence, and (iii) perform the processes of text integration to establish by inference, if
necessary, the "mathematical connections" between sets which (iv) in turn lead to arithmetic actions to
perform on those sets.

The implementation of these 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 predicates of objects whereas in
word problems, quantified noun groups must be abstracted to the point where the numbers are the objects
of interest (Nesher and 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" (DeCorte and Verschaffel, 1985), understanding "textual presuppositions" (Kintsch and
Greeno, 1985), and possessing the "mathematics register" (Spinos, Rhodes, Dale and Crandall, 1988).

The next two sections present two interacting components of the computer model which (i) read
individual sentences in word problems and (ii) perform text integration between sentences in order to
establish the relationship between sets. For each component, detailed examples are provided to define

Figure 1.
The fundamental transitions between representations facing the word problem solver
their respective, often overlapping, roles in the comprehension process.

3.3. EDUCE - The Sentence Reader

EDUCE\(^1\) is an expectation-based conceptual analyzer (Burns, 1993; LeBlanc & Russell, 1989) adapted from the work of Schank and Riesbeck (1981) and Birnbaum and Selfridge (1981) and developed according to principles found to be distinct for word-problem-solving. For each sentence in a word problem, the conceptual-analysis-based parser builds a number of conceptual elements and combines those conceptual elements into a canonical representation of the sentence, i.e., a unique conceptual representation of its meaning.\(^2\) Unlike syntactic parsers which first construct syntactic structures and then attach semantic meanings to those structures, conceptual analysis is semantically driven and uses both syntactic and semantic information as it proceeds from the lexical to the conceptual representation of a sentence.

In "reading" a sentence in an arithmetic word problem, EDUCE builds its structures on the basis of semantic and syntactic "expectations" of concepts to follow, as specified by words as they are encountered. As it "reads" each sentence in an arithmetic word problem, EDUCE handles a number of language features inherent in word problems, including: (i) quantities (e.g., 3, three); (ii) noun compounds (e.g., soda cans); (iii) pronominal reference (e.g., she); (iv) time sequence (e.g., then, in the beginning) (v) set partitions (e.g., [by ownership: "Dick and Jane have"], [by type of object: "8 cats and dogs"]); (vi) ellipsis (e.g., "Ed has 3 cans. He found 2 more." That is, 2 more cans) and (vii) reference to previous sets (e.g., 4 of them).

The next two sections provide detailed examples of EDUCE's parsing capabilities. The first example highlights the processing involved in expectation-based parsing while the second example focuses on specific word problem features in EDUCE. The final section presents an example of how the design of

\(^1\)EDUCE stands for Explaining Discourse Understanding with Conceptual Expectations.

\(^2\) Representations of text at this level are of the type created by Schank (1975), Wilks (1973) or Norman and Rumelhart (1975), which use semantic and conceptual primitives to represent paraphrases canonically and serve as a medium for translation. The design of the model’s parser discussed in this section assumes an adaptation of Schank’s conceptual dependency representations.
EDUCE's lexicon is providing new insight into the role of specific mathematical words.

3.3.1. A Parsing Example

A sample parse of the sentence,

Jake has 5 soda cans.

can help clarify the processes in expectation-based analysis. This example sentence is typical of those commonly found in an arithmetic word problem. The primary intent of this example is to focus attention on the many cognitive processes that are potentially involved in reading and "conceptualizing" even a simple sentence. In addition to parsing each sentence in a word problem, EDUCE monitors the linguistic problem-solving requirements which occur while reading individual sentences.

Each word in the sentence is read one at a time starting from left to right. Upon reading the first word, "Jake," EDUCE recognizes this word as a name and performs a referential search to see if Jake has been previously mentioned. If Jake had been previously mentioned then the already existing concept for Jake is accessed. Assuming for this example that this is the first mention of Jake, EDUCE accesses the word's lexical entry\(^3\) and loads its conceptual meaning into short-term memory (STM)\(^4\):  

CONCEPTS: (1) PERSON -- (physical (+human) (reference: Jake))  

REQUESTS: nil

There are no expectations or requests generated by the word Jake, i.e., 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 STM (2) along with its associated requests:

---

\(^3\) EDUCE does not model the fundamental reading processes of letter recognition like the model of Liberge and Samuels (1974). In EDUCE, a word is "read" as a single unit and lexical access activates the word's conceptual meaning and associated conceptual expectations (i.e., what other concepts this concept expects to find during the processing of the sentence).

\(^4\) The convention of referring to short-term memory (STM) and long-term memory (LTM) as "locations" serves as a convenient method of speaking of "level of activation." LTM represents those concepts which have presumably "decayed" to a state whereby reactivation is necessary in order to reference them.
CONCEPTS: (1) PERSON -- (physical (+human) (reference: Jake))
(2) STATE: possession
OBJECT: ?
POSESSOR: ?

REQUESTS: (i) Who possesses? [search for a human earlier in the 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 since a search finds the conceptual representation of Jake in STM (i.e., Jake's conceptualization is +human) and thus the (1) Jake-concept is merged into the POSSESSOR-slot of the (2) STATE-possession-concept. The second request (ii) is unsatisfied since no conceptual-object currently resides in STM. Having read and completed the processing for the first two words, "Jake has." EDUCE has one concept and one outstanding request in STM:

CONCEPTS: (1) STATE: possession
OBJECT: ?
POSESSOR: (physical (+human) (reference: Jake))

REQUESTS: (i) What is possessed?

Table 1 summarizes the results of monitoring the linguistic processing which has occurred during the comprehension of these first two words, "Jake has." The left column lists some of the linguistic variables (or tacit reading tasks) that EDUCE monitors as it parses a sentence and the right column lists the current measure of the respective variable. For example, since there was one (1) concept in STM for the word "Jake" and two (2) concepts in STM for the word "has," the maximum number of STM concepts is 2 and the average number of STM concepts is 1.5 \( \frac{(1+2)}{(2 \text{ words})} \). Since "Jake" posted no requests (0) and "has" posted two (2) requests, the maximum number of conceptual requests which have been posted for any word is two and the average number of requests posted is 1.0 \( \frac{(0+2)}{2} \). Also, since the Jake-concept was merged into the POSSESSOR slot of the STATE-concept, one (1) conceptual merge has occurred.
Table 1.
Summary of EDUCE’s Processing for “Jake has”

<table>
<thead>
<tr>
<th>Summary of Linguistic Processing Jake has...</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum # of STM concepts</td>
<td>2</td>
</tr>
<tr>
<td>average # of STM concepts</td>
<td>1.5</td>
</tr>
<tr>
<td>maximum # of requests posted</td>
<td>2</td>
</tr>
<tr>
<td>average # of requests posted</td>
<td>1.0</td>
</tr>
<tr>
<td>maximum # of requests fired</td>
<td>1</td>
</tr>
<tr>
<td>average # of requests fired</td>
<td>0.5</td>
</tr>
<tr>
<td>number of merged concepts</td>
<td>1</td>
</tr>
</tbody>
</table>

The next three words, “5 soda cans,” form a noun group and are eventually merged into an “object-concept” with a quantity of five. This noun group is especially complicated to parse since it contains a noun-compound, “soda cans.”5 In short, the “soda cans” compound is merged into one object-concept where the function of the can-object is to contain soda. The quantifier “5” then adds a quantification-slot to the can-object. Hence, “5 soda cans” is merged into one object-concept. 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 STATE-concept. A final conceptualization for the sentence resides in STM:

CONCEPTS: (1) STATE: possession
OBJECT: (physical (-animate) (function: contain(object: soda)) (quantity: 5))
POSSESSOR: (physical (+human) (reference: Jake))

5 Although many noun compounds require specific parsing strategies and extra processing for EDUCE, it does not imply that children face similar challenges when encountering noun compounds. For example, “soda cans” is so common that most children probably understand “soda cans” as an idiom and therefore read it as one word, e.g., “soda-cans.” EDUCE has the flexibility to handle noun compounds as idioms or perform more complex strategies to deduce that “soda cans” is “a physical object which has the function of containing soda.”
Table 2 summarizes the linguistic processing which occurred in order for EDUCE to arrive at such a conceptual understanding of the entire sentence.

3.3.2. A “Mathematical Reading” Example

This example highlights some of the word-problem specific capabilities of EDUCE in the context of parsing the second sentence of the arithmetic 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?

A sample parse of the sentence, “Jacob has 3 of them,” with a focus on the relevant phrase “3 of them” can help clarify how EDUCE initiates the process of text integration. At this point, it is assumed that EDUCE has read and interpreted the first sentence as “a set of 8 belonging to Jacob and Kathy.”

Each word in the second sentence is read one at a time starting from left to right. After reading and completing the processing for the first two words in the second sentence, “*Jacob has,*” EDUCE has one concept and one outstanding request in short-term memory (STM):

<table>
<thead>
<tr>
<th>Summary of Linguistic Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Jake has 5 soda cans.</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable</th>
<th>Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>maximum # of STM concepts</td>
<td>4</td>
</tr>
<tr>
<td>average # of STM concepts</td>
<td>3.2</td>
</tr>
<tr>
<td>maximum # of requests posted</td>
<td>5</td>
</tr>
<tr>
<td>average # of requests posted</td>
<td>1.7</td>
</tr>
<tr>
<td>maximum # of requests fired</td>
<td>2</td>
</tr>
<tr>
<td>average # of requests fired</td>
<td>0.5</td>
</tr>
<tr>
<td>number of merged concepts</td>
<td>4</td>
</tr>
</tbody>
</table>
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 (i) enter noun group mode; (ii) instantiate a conceptual quantity; and (iii) 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 conceptual referent to follow (currently unknown) is the whole which 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 "3 of 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 which 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 STATE-concept. A final conceptualization for the sentence resides in STM:

CONCEPTS: (I) STATE: possession
POSSESSOR: (physical (+human) (DEFINITE reference: Jacob))
OBJECT: (physical (-animate) (function: contain(object: soda))
(quantity: 3) (PART-OF: "a previous set containing this object")

Of specific importance in this example 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 in the next section, such explicit wordings and their associated representations facilitate the processes of text integration. As a counter-example, a more traditional wording of the second sentence in this type of problem is: "Jacob has 3 soda cans." In this case, EDUCE encounters no explicit set reference and thus would not be able to represent any con-
connection between the two sentences.

3.3.3. Starting With the Words

As stated previously, the design of this word problem solving model 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. The decision was based in part on the fact that no previous model of arithmetic word problem solving included the processes of reading individual sentences. More specifically, the impact of slight changes in problem wording and the role of certain "mathematical" words and phrases was not addressed. As shown in the previous parsing example with the phrase "of them," EDUCE is sensitive to changes in wording which facilitates text integration. In addition, the design of EDUCE's lexicon provides a research tool for investigating the often subtle contributions of certain words to the overall meaning of a sentence. The next section presents a detailed look at how EDUCE defines the role of "mathematical" words and how slight changes in the meaning of those words can affect problem solving performance.

3.3.3.1. At the Lexical Level

In a complete processing model of word problem solving, the development of the lexicon is critical to define the role of individual words precisely, especially the role of certain "mathematical" words such as quantifiers (e.g., some) and relational terms (e.g., more). In EDUCE, the lexicon contains both syntactic (e.g., part of speech) as well as semantic (e.g., type of conceptual act) information for each word. In the context of a complete model, the semantic component for each word defines the contribution of individual words toward the overall meaning of a sentence and ultimately toward the underlying relationships between sets. As important, the design of the lexicon allows for flexible updates to the lexicon to reflect, for example, children's common misunderstanding of certain words and phrases.  

6 The complete design of EDUCE, which incorporates tools for its flexible use and modification, is beyond the scope of this work. Interested readers are referred to Burns (1993).
3.3.3.2. Altogether

The word "altogether" provides an example of how EDUCE provides an ability to investigate the role of specific words. Figure 2 shows a simplified version of EDUCE's lexical entry for the word "altogether." The lexical entry can be interpreted as follows. Quantified object concepts can be partitioned by ownership (e.g., "John and Mary have 8 marbles."), type of object (e.g., "John has 8 cats and dogs."), and other qualifiers such as color (e.g., "John has 8 red and blue marbles."). Since quantified noun groups containing the word "and" can also partition a group of objects, (e.g., "8 red and blue marbles"). it is possible that the word altogether does not generate a partition but only reinforces a previously established partition. Upon parsing the word "altogether," EDUCE searches working memory for a quantified object in the current sentence and checks to see if that conceptual object (e.g., 8 marbles) is already understood to be partitioned, that is, whether the entire collection of objects is made up of distinct parts. If the object concept is not already partitioned (ELSE), EDUCE marks the object as partitioned, although it does not yet know how to partition, that is, the partition-type (e.g., ownership, type of object, color) is unknown. If no partition-type is known, EDUCE checks whether the sentence has two owners in possession of this object; if so, the partition-type is set to OWNER. Otherwise, EDUCE can not determine the partition-type from the information in the sentence, although it does know that it is partitioned. This situation is possible if EDUCE reads a sentence such as: "John has 8 marbles altogether." In this case, "altogether" implies that John's 8 marbles are partitioned, although it is not clear from this sentence exactly how they will be partitioned. If the next sentence reads: "3 of them are blue," the text integration component must infer that John's 8 marbles are partitioned by color.

7 In previous models of arithmetic word problem solving, the role of the word altogether is lost or trivialized. In the models of Riley et al. (1983), Kintsch and Greeno (1985) and Cummins et al. (1988), a sentence such as "John and Mary have 8 marbles altogether" is propositionalized as HAVE-ALTOGETHER(John, Mary, 8) prior to the simulation. In the clueword parsing model of Briars and Larkin (1984), arithmetic actions are instantiated based on rules of the form: "IF the word altogether appears and there are two previous sets THEN join the two sets into one set."
IF (object-concept is already partitioned)

- "altogether" reinforces the partition

ELSE

- make a partition in the object-concept, i.e., partition-type = 2
- IF current sentence ACT=POSSESSION and there are 2 owners
  - set partition-type to owner, i.e., partition-type = OWNER
- ELSE
  - do nothing (partition-type will have to be inferred later)

Figure 2.
A simplified version of EDUCE's lexical entry for the word "altogether"

In short, "altogether" implies that a group of objects is partitioned in some way, although the word alone does not imply (i) how the set is partitioned or (ii) that the model should use addition. These two, often subtle points become evident once the lexical entry for "altogether" is constructed.

3.4. Summary of EDUCE

EDUCE is a sentence-level parser which reads each sentence in a word problem and constructs an unambiguous representation of the meaning of that sentence. Because EDUCE reads in a word by word fashion, the lexicon depicts the contribution of individual words, specifically, how mathematical terms affect the meaning of a sentence. In addition, this component highlights some of the critical text comprehension tasks which occur at the sentence level (e.g., pronominal reference, time sequence, ellipsis). More important, EDUCE is sensitive to changes in wording which can facilitate the integration between sentences. As discussed in the next section, specific links between sentences as established by EDUCE reduce the amount of inferencing that is required of the text integration component.
3.5. SELAH - Text Integration

SELAH\(^4\) is a computer model which accepts EDUCE's conceptualized representations of sentences as input and attempts to integrate the current input with previously conceptualized sentences. A link between conceptualized sets is either inferred by SELAH or is indicated by a reference to a previous set in the conceptualization produced by EDUCE. Integrated sets in turn produce a representation of the situation which enables the activation of an arithmetic action, such as joining two sets together or separating one set from another set.

SELAH is currently implemented as a "bottom-up" problem solver. SELAH completely determines the role of each sentence after it is read by EDUCE and then the entire reading and text integration process continues to the next sentence. Bottom-up processing is opposed to a more global, expert-like, "top-down" strategy that potentially involves an instantiated schema in memory for each particular type of problem and/or an associated plan to look for quantities which can fill slots in that schema. Although SELAH has been designed so that it could function in a more expert-like top-down fashion, the focus of the current implementation is a model of the young reader or, at least, a novice word problem solver.\(^7\)

More specifically, the focus is to model the necessary text comprehension and logico-mathematical requirements of problem solution. For example, some problems require higher memory loads than other problems. While it is possible to set a limit on the number of concepts which can be stored in working memory (cf. Fletcher, 1986), SELAH, as described below, solves each problem with an unlimited memory capacity. As discussed in the experiments in the next chapter, one theoretical interest is the total demand on memory that each problem requires when read in a bottom-up fashion. In short, SELAH simu-

\(^4\)Selah (pronounced see-lah in this context) is an ancient Hebrew musical term whose original meaning is largely unknown. It appears frequently in the right-hand margin of the Book of Psalms. One contemporary view considers it to mean: "Pause and calmly think about it."

\(^7\) At this stage, SELAH is not a developmental model in the sense of Riley et al. (1988) and Briars and Larkin (1984). That is, in the current implementation, SELAH does not alternate between varying levels of expertise. Because SELAH solves all of the problems in the benchmark set, it exhibits expert-like capabilities (e.g., performing a relatively large number of inferences) on some problems but only rudimentary capabilities on the "easiest" problems.
lates a problem solver who reads each sentence only once and attempts to maintain the sets and their relationships in memory until deciding on an appropriate arithmetic action which can lead to a solution.\footnote{The current implementation of SELAH does not simulate solutions which use manipulatives such as blocks on the table (cf. the CHIPS model of Birt & Larkin, 1984). In a sense, children who use manipulatives are closer to a strict definition of bottom-up problem solving; that is, for each sentence that is read the blocks are moved. The focus of SELAH is to model the total text comprehension requirements that novices experience as they solve these problems, including the cognitive demands of remembering previous sets rather than offloading those requirements by depending on physical aids.}

### 3.5.1. A Text Integration Example

Some of SELAH's text integration processes are explained in 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?

As each sentence is parsed, EDUCE passes its conceptualization of that sentence 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 integration is possible, since 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:

<table>
<thead>
<tr>
<th>LTM-(1) STATE: possession</th>
</tr>
</thead>
<tbody>
<tr>
<td>POSSESSOR: (Jacob Kathy)</td>
</tr>
<tr>
<td>SET:</td>
</tr>
<tr>
<td>OBJECT: (physical (-animate) (function: contain(object: soda)))</td>
</tr>
<tr>
<td>QUANTITY: 8</td>
</tr>
</tbody>
</table>

At this point, control returns to EDUCE and the second sentence is parsed. As shown in a previous parsing example, EDUCE generates a PART-OF slot from the "of them" phrase in the second sentence 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 sen-
tence. SELAH searches LTM for the set which 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, i.e., the set in LTM-(2) is PART-OF the set in LTM-(1):

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 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 mental 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 must first construct procedural or action-oriented representations in order to arrive at an arithmetic operator. While some problems include explicit actions (e.g., giving) which facilitate children's success (Briars & Larkin, 1984), many word problems do not contain action language and require additional processing to infer a

---

11 In this example, the reference from the word "them" insures 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).
procedural interpretation of statically expressed relationships. In the current example, SELAH infers a *separating-from* interpretation of a static *part-of* relationship between two sets.

The third sentence involves a similar link between the third sentence and the first, i.e., 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 short, SELAH’s current representation is: LTM-2 SEPARATED-FROM LTM-1 resulting in LTM-3. 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 which led to this current representation.

---

**Table 3.**

Summary of SELAH’s processing on an “of them” problem

| Jacob and Kathy have 8 soda cans. Jacob has 3 of them. The rest are Kathy’s. | Sentence |
| --- | --- | --- | --- | --- |
| **Processing Summary** | 1 | 2 | 3 | Average |
| make set | explicit part-of; infer SEP-FROM | explicit part-of; reinforce SEP-FROM; explicit result of SEP-FROM |
| **Concepts in Memory** | [J&K’s 8] | [SEP J’s 3] | [SEP J’s 3] | [FR J&K’s 8] | [RE K’s?] |
| # of concepts | 1 | 2 | 3 | 2.00 |
| # of inferences | 0 | 1 | 0 | 0.33 |
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, e.g.,

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

**3.6. The SELAH "Algorithm"**

The SELAH component is an attempt to model the independent and yet overlapping processes of text integration and mathematical problem solving. Although logico-mathematical knowledge (for example, two parts make up a whole) is important, SELAH’s emphasis is on specifying (i) the possible stages of representation that are traversed and (ii) the "amount of cognitive processing" that is required during the traversal of these stages (e.g., the average number of sets that must be remembered).

Given an unambiguous natural language interpretation of a word problem, SELAH must bridge the gap between what is being stated and the correct mathematical relationships. While an "expert" problem solver may make the transition simultaneously as the problem is read, SELAH follows a more "novice-like" sequential strategy. A unique feature of SELAH’s sequential strategy is that all problems are eventually understood in a procedural framework, including those problems which do not explicitly include action-oriented language. The instantiated procedure is either the arithmetic action of *Joining* two sets together or *Separating* one set from another set.

As stated previously, SELAH operates on conceptualized representations of sentences as produced by EDUCE. SELAH makes six (6) specific distinctions as it decides how to handle EDUCE’s output. SELAH makes an initial distinction between the conceptual categories of (1) actions (e.g., transferring objects between people) and states (e.g., people possessing objects), in line with conceptual dependency
theory (Schank & Abelson, 1977). Static conceptualizations which represent states are further delineated according to the direct or implied relationships included within a state description, e.g., (2) "have altogether," (3) "have some of them," (4) "have less than," (5) "have more than," or simply (6) "have."

Figure 3 shows the SELAH algorithm which performs this initial categorization. The next sections present SELAH’s capabilities in each of these six areas in the context of representing sets, establishing

```plaintext
IF (new_sentence_concept_is_an_ACTION?)
   (1) (handle_action_sentence_concept)
ELSE IF (new_sentence_concept_is_a_STATE?)
   IF (STATE_contains_a_PARTITION?)
      (2) (handle_PARTITION_sentence)
   ELSE IF (STATE_contains_a_PART_OF?)
      (3) (handle_PARTOF_sentence)
   ELSE IF (STATE_contains_RELATIONAL_language)
      IF (LESS_THAN)
         (4) (handle_less_than_sentence)
      ELSE (MORE_THAN)
         (5) (handle_more_than_sentence)
   OTHERWISE
      (6) (handle_ISOLATED_SET_POSSESSION)
```

Figure 3.

SELAH's initial distinctions between conceptual representation components produced by EDUCE

---

12 SELAH’s knowledge of how to act on certain types of conceptual representations is currently implemented in a procedural (if-then-else) fashion, as opposed to a declarative (rule-based, expert-system) approach. The procedural implementation in this prototype model has provided critical insights to the distinction between knowledge representation and control of that knowledge that is essential for expert system development and yet was not obvious at the start of a project of this magnitude.
connections between sets and using those connections to instantiate a procedural interpretation of the underlying relationships.

3.6.1. Making Isolated Sets

If the conceptualization from EDUCE contains a conceptual state with no implicit or explicit reference to other sets, SELAH recognizes this as an isolated set. Before any set is stored in SELAH’s working memory, the focus of the conceptual representation is changed to a more mathematical focus on the *quantity*, *type* and *owner* of the collection of objects. This change in focus simulates the continued transition to increasingly abstract representations, where ultimately SELAH focuses on the quantities. For example, the representation of a set in SELAH’s memory does not indicate how that set was described in natural language or how it may have been formed, for example, as the result of a previous transfer.

SELAH’s set representation:

```
LTM SET:
  QUANTITY: 3
  OBJECT: "gumball"
  OWNER: Zac
```

may be the result of a number of conceptualizations such as “Zac <POSSESSES> [3 gumballs]” or “Sarah <TRANSFERRED> [3 gumballs] to Zac.” Of particular importance is that SELAH remembers the set of “Zac’s 3 gumballs.”

3.6.2. Handling Conceptual Actions

Conceptual actions play a unique role in SELAH’s processing because of the connection between, for example, a transfer of possession, and the instantiation of an arithmetic action, such as *Join* or *Separate-From*. Depending on the context, SELAH recognizes conceptual transfers as one of two situations: a *transfer-IN* to an already existing set or a *transfer-OUT* of an already existing set. SELAH recognizes a *transfer-IN* situation when the recipient of a transfer already has a set in his possession, that is, a set previously stored in memory. The “recognition” is triggered by four sequential conditions: (i) a
reference to the recipient of the transfer is via pronominal or direct reference; (ii) a set is found in
memory with the same owner as the recipient (reference); (iii) the set in memory is in the recipient’s pos-
session prior to the current transfer; and (iv) the objects in both the previous set and the transferred set
are the same. If these conditions are met, SELAH “automatically” (that is, without having to make an
inference) instantiates the arithmetic action Join. That is, SELAH reasons that “a transfer of a certain
number of objects to a person who is already in possession of a certain number of those same objects
instantiates the “mathematical” situation where two amounts are joined together.” Because the transfer-
IN is explicit in the conceptualization produced by the reading component, SELAH is not required to
infer that the transferred amount should be joined to the previous amount.

Recognition of a transfer-OUT situation is similar to transfer-IN except that the reference and
search through memory is directed toward the source of the transfer and the instantiated action is to
separate the transferred quantity from the previous amount.

The following example provides an annotated script of SELAH’s output as it handles a transfer
situation, specifically in the context of the following word problem:

Zac had 5 gumballs. Then Sarah gave him 3 gumballs.
How many gumballs does Zac have now?

The first sentence results in SELAH making a set of [Zac’s 5 gumballs] and storing that set in memory.


-----------------------------
MAKE SET (1): ZAC 5
tense = PAST time = 1


The second sentence involves a “transfer of possession” act. SELAH accepts as input EDUCE’s
representation of this action:
/* EDUCE output of sentence 2 */
ACT: transfer
OBJECT: 3 gumballs
FROM: Sarah
TO: Zac (via direct reference)

SELAH recognizes this action as a transfer-IN situation since the recipient of the transferred amount, Zac, has a previous set in memory. The explicit transfer-IN situation instantiates the arithmetic action Join, that is, [Join the transferred amount to the previous amount] as shown in SELAH’s output:

&;&;&;&;&;&;&;&;&;&;&; SENTENCE #2 &;&;&;&;&;&;&;&;&;&;&;&
-----------------------------------------------
Sarah -TRANS(3)-> him
(tense: PAST time: 2)
-----------------------------------------------

Pronominal Reference: ZAC
(Searching Memory ...)

REFERENCE(ZAC) HAS A PREVIOUS SET

REFERENCE(ZAC) IS RECIPIENT
of amount transferred
-----------------------------------------------
MAKE SET (2): ZAC 3
tense = PAST time = 2
-----------------------------------------------
(objects in both sets are "same")

TRANSFER-IN Situation INSTANTIATED
**************************************

INSTANTIATE ACTION JOIN
*************************
JOIN set(2) TO set(1)
*************************

&;&;&;&;&;&;&;&;&;&;& END SENTENCE #2 &;&;&;&;&;&;&;&;&;&

At this point, that is after SELAH has processed the second sentence, SELAH’s working memory contains two sets and the arithmetic action Join. As SELAH solves each problem, it monitors the amount

---

1 For any number of sentences that are identical in meaning, regardless of language, EDUCE produces only one representation. Thus, EDUCE generates this "transfer of possession" representation from a number of possible natural language sentences, for example: "Sarah gave him 3 gumballs," "Zac got 3 gumballs from Sarah," or perhaps the more realistic scenario, "Zac took 3 of Sarah’s gumballs."
of units or "concepts" in working memory at the end of the sentence. 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 5)], whereas [Join 3 & some] is counted as three units: [(Join) (3) (some)]. The rationale for reducing the memory load in these cases through chunking is based on the fact that all the necessary information for a solution is now available, assuming of course that the question to follow requests the amount of the resulting set. If children were using manipulatives to solve this problem, three blocks would be joined to the previous set of five blocks and the resulting physical representation of the situation would collapse to one unit, a set of eight blocks. Because SELAH relies on mental representation of sets and actions, the assumption is that a memory load of joining two sets with known quantities in one's head (2 units) is slightly greater than doing it with manipulatives (1 unit) but is less than joining two sets where one set has an unknown quantity (3 units). In this example, SELAH has two units in working memory: [Join (3 5)] after processing the second sentence.

The third sentence requests the amount that "Zac has now." SELAH creates a new set with an unknown amount and marks this set as the set in question. At this point, SELAH proceeds to a selection of a counting strategy which mirrors the current arithmetic action. Counting strategies represent the highest level of abstraction in SELAH's problem solving process, as only quantities and the arithmetic action are required. In this example, "Join 3 to 5" is mapped onto the Counting On From First counting strategy.  

---

14 SELAH does not simulate more advanced versions of the Counting On From First strategy such as Counting On From Larger that many children learn to use as they become sensitive to counting efficiency. In addition, because SELAH is simulating a novice problem solver, it does not rely on number facts, e.g., "3+5 is 8."
Attempt to Answer Question

CURRENT ACTION is JOIN

Known Quantities involved in this action are:

\[ 3 + 5 = ? \]

COUNTING ON FROM FIRST

\[ \]

3
4 .... (that's 1)
5 .... (that's 2)
6 .... (that's 3)
7 .... (that's 4)
8 .... (that's 5)

The answer is 8!!

Table 4 summarizes some of SELAH's processing in this example. Note that this problem does not

Table 4.
Summary of SELAH's processing on an action problem

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence 1</th>
<th>Sentence 2</th>
<th>Sentence 3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts in Memory</td>
<td>[Zac's 5]</td>
<td>[JOIN Sarah's 3]</td>
<td>[JOIN Sarah's 3]</td>
<td>2.00</td>
</tr>
<tr>
<td></td>
<td>[TO Zac's 5]</td>
<td>[TO Zac's 5]</td>
<td>[RE Zac's?]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>

Zac had 5 gumballs. Then Sarah gave him 3 gumballs. How many gumballs does Zac have now?
require SELAH to make any inferences. For example, no inference was needed to instantiate the arithmetic action, Join, since the join action was explicit in the conceptual transfer-IN representation. The average memory load, 2.00 units per sentence, is also relatively light. As will become evident from the empirical evidence in the next chapter, a small number of inferences and a light memory load lead to a prediction that most children will be able to solve problems of this type. Although action language generally reduces the amount of inferencing that SELAH performs, problems with explicit actions do not always involve low measures in SELAH's processing. Chapters 4 and 5 outline the impact of such differences in more detail. Appendix B provides annotated scripts for other action problems that SELAH solves.

3.6.3. Handling Non-Relational Static Relationships

Word problems with static relationships are defined as those problems which describe the combination and comparison of sets. These problems use "mathematical" terms such as altogether to describe the combination or association of disjoint sets and phrases such as more than to describe the relational comparison between disjoint sets. A distinguishing feature of SELAH's processing on static problems, as opposed to the action-oriented problems in the previous section, is that SELAH has no explicit cue for the instantiation of an arithmetic action. The model must infer arithmetic actions such as Join based on the processes of text integration. Because relational language (e.g., more than, less than) is exceptionally difficult for most children, static-relational problems are discussed separately in the next section. This section focuses on the static-nonrelational problems, typically referred to as Combine problems.

Solving Combine problems involves an ability to associate or disassociate (depending on how the problem is presented) two sets. Because these problems do not involve significant actions, SELAH relies more heavily on part-part-whole knowledge, that is, the joining of two parts (subsets) results in a third whole (super) set. The emphasis on the joining action represents a critical distinction between the way SELAH solves these problems and the abstract, expert-like part-whole problem solving knowledge in
previous computer simulations (e.g., Kintsch & Greeno, 1985). Specifically, EDUCE initially interprets the phrases “in all” and “altogether” as describing a set that “has been grouped” according to some feature (e.g., by ownership). SELAH, in turn, interprets this static grouping in a procedural way. The grouped set is considered to be the result of a hypothetical arithmetic join action. The action is hypothetical because the problem text does not explicitly state the action. For example, EDUCE interprets the sentence: “Kay and Mary have 8 dolls altogether” as a set which can be grouped (and therefore partitioned) by ownership. SELAH infers this grouping by ownership to be the result of joining the dolls that Kay owns with the dolls that Mary owns. In short, SELAH interprets a partitioned set to highlight the result of an unmentioned Join.

At this point, it is essential to understand that when SELAH infers a Join action, it does not mean that SELAH has decided to use the arithmetic operator of addition. SELAH knows that two sets must be joined to form a resulting set, however, the sets are not blindly chosen to be those that have known quantities. SELAH’s choice of an arithmetic operator is not decided until a complete representation of the relationships in the problem is achieved. For example, if the two sets to be joined have known quantities (join 5 to 3), then SELAH will eventually choose to add the two quantities with the Counting on From First strategy (3 + 5 = ?). However, if only one of the parts has a known quantity and the resulting amount of the join is known, SELAH’s interpretation is that some quantity must be joined to 3 to get a result of 8. Here, SELAH resorts to a more complex Counting Up From Given strategy (3 + ? = 8). This focus on the relationship among quantities in the join action is in contrast to a “clueword” strategy that is often exhibited in young children, that is, “altogether means use addition.” Children who rely on a clueword strategy often answer “11” to problems of the form (join some to 3 to get 8). The difference is a focus on the role of sets vs. a focus on the quantities of sets.

The following example provides an annotated script of SELAH’s output as it handles a non-relational static problem, specifically in the context of the following word problem:
Zac has 5 gumballs. Sarah has some gumballs. Zac and Sarah have 8 gumballs altogether. How many gumballs does Sarah have?

The first two sentences result in SELAH making a set of [Zac’s 5 gumballs], making a set of [Sarah’s some gumballs] and storing those two sets in memory.\textsuperscript{15}

\begin{verbatim}
&\\&\\\&\\&\\&\\&\\&\\ SENTENCE #1 &\\&\\&\\&\\&\\&\\&\\
MAKE SET (1): ZAC 5
tense = PRESENT  time = 1

&\\&\\&\\\&\\&\\&\\&\\&\\&\\&\\&\\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\\&\end{verbatim}

The third sentence involves a static description of a set jointly owned by Zac and Sarah. SELAH accepts as input EDUCE's representation of this non-relational sentence:

```
/* EDUCE output of sentence 3 */
STATE: possession
OWNER:  (reference-Zac & reference-Sarah)
OBJECT:  8 gumballs
GROUPED-BY: ownership
```

SELAH recognizes this grouped set of objects as "a set whose objects are the result of joining the objects from the owners Zac and Sarah." Because EDUCE's representation only implies the result of an action, SELAH must infer that a Join action hypothetically occurred. In addition, SELAH searches its working memory and confirms that Zac and Sarah (the owners of this jointly-owned set) are references to the owners of two previous sets: set(1) and set(2). SELAH infers that these two previous sets, [Zac’s 5] and [Sarah’s some], are legitimate parts of this jointly-owned set of 8, where "legitimate" concerns the constraints checks: (i) the type of objects in the referenced set is the "same" (or like-type) as the objects in

\textsuperscript{15} In this sample run, EDUCE correctly understands the meaning of the word "some" to be a quantifier, that is, "some gumballs" is understood like "5 gumballs." As noted by Cummins et al. (1988), the word "some" is often misinterpreted by children to be an adjective rather than a quantifier, that is "some gumballs" is understood like "red gumballs." In this example, EDUCE interprets "some" as a quantifier so gumballs is represented as an object with an unknown amount. SELAH represents this as a set with an unknown cardinality.
the referring set and (ii) the quantity in the referenced set is either unknown or less than the amount in the referring set.

(Pronominal Reference: ZAC
Pronominal Reference: SARAH

(Searching memory ...)
REFERENCE(ZAC) HAS A PREVIOUS SET
INFER:
ZAC’s 5 PART-OF (ZAC and SARAH)’s 8

(Searching memory ...)
REFERENCE(SARAH) HAS A PREVIOUS SET
INFER:
SARAH’s some PART-OF (ZAC and SARAH)’s 8

!!!!!!!! INFERRED !!!!
Procedural JOIN situation

INSTANTIATE ACTION JOIN

MAKE SET (3): (ZAC & SARAH) 8
tense = PRESENT time = 3

At this point, that is after SELAH has processed the third sentence, SELAH’s working memory contains three sets and the arithmetic action Join, for a total of four memory units.

The fourth sentence requests the number of Sarah’s gumballs. SELAH notes the reference to a previous set owned by Sarah and marks this set as the set in question. Because one of the quantities to be joined is unknown, SELAH converts the Join action to an Add-On action, that is, add some on to 5 until arriving at 8. Given an Add-On action, SELAH chooses the counting strategy “Counting Up From Given.” In this example, the Counting Up From Given strategy uses a forward counting sequence starting with 5 and continues until 8 is reached. The answer is the number of counting words in the sequence.
Table 5 summarizes some of SELAH’s processing as it solved this example. Note that SELAH must make three inferences while maintaining a relatively high load on working memory. Appendix B provides annotated scripts for other static, non-relational problems that SELAH solves.

3.6.4. Handling Relational Static Relationships

Solving problems with relational language (that is, Compare problems) involves an ability to consider two disjoint sets with a focus on the difference in cardinalities between those two sets. Like other static problems (e.g., Combine problems), relational (Compare) problems do not involve significant
actions, so SELAH must infer an action to transform a static description into a procedural situation. On the other hand, a feature which distinguishes relational problems from other static problems is that the difference between two sets is not easily recognized as a "set," as pointed out by Riley and Greeno (1988):

In the models we used (Riley and Greeno models), the representation of a comparison between sets has the same form as the representation of set unions in the part-whole schema. An important difference exists, however, between the two cases. The union of sets is a set, but the difference between two sets is not a set. This means that the numbers involved in set unions can be understood as the cardinalities of individual sets, but the numbers involved in set differences must be understood as a relations between sets or between the cardinalities of two sets (p. 85).

Another unique feature of relational word problems is that there are two ways to describe the difference between two sets: "x is more than y" or "y is fewer than x." The descriptions are of course a matter of focus. If the difference relation is expressed with a focus on the larger set, the difference is described as being more than; with a focus on the smaller set, the difference is described as fewer than.
The examples to follow highlight how SELAH is providing new insight into the tacit processing that is required when novice problem solvers interpret relational expressions. The first annotated script of SELAH’s output reveals how SELAH is able to interpret more than problems in a non-relational fashion. The second example shows how SELAH must resort to relational knowledge in order to solve fewer than problems.

3.6.4.1. “More” and “More Than”

An understanding of how SELAH handles relational problems begins with a look at how EDUCE interprets and represents the individual words in relational expressions. Because EDUCE reads sentences in a word by word fashion, each word has an independent lexical meaning. The “meaning” of each word can vary depending on the level of expertise being simulated by the reading model. Specific to the model’s current focus on a young reader, the contribution of the word “more” to the meaning of a sentence is that the quantity in the current sentence is “in addition to” some other quantity. This interpretation combines (i) the qualitative sense of the use of the word that young children learn quite early, for example, “I want more cake,” as well as (ii) a quantitative sense of the word which implies a join action. For example, given the sentences: “Betsy had 3 shells. Bill gave Betsy 2 more shells,” EDUCE interprets the second sentence to mean that “Bill gave Betsy 2 shells which are in addition to [some previous unmentioned amount].” That is, the 2 given to Betsy are “an increase over and above [some previous amount].” Of importance is that the model’s lexical definition of the word “more” always implies a referent, i.e., in addition to [what]? In this example, because the second sentence does not explicitly describe a previous amount, EDUCE’s representation can not include a specific referent. During the process of text integration, SELAH notices the implied reference to a previous amount: [some previous amount], and infers that Betsy’s original [3 shells] is the missing referent. Following the integration of the two sentences, SELAH interprets the second sentence to mean “Bill gave Betsy 2 shells in addition to [the 3 shells she previously had].”
On the other hand, sentences containing the relational expression "more than" do indicate a specific referent to a previous amount. For example, given the specific problem:

Betsy has 3 shells. Bill has 2 more shells than Betsy. How many shells does Bill have?

EDUCE interprets the second sentence to mean that "Bill has 2 shells in addition to [the number of shells that Betsy has]."¹⁶ That is, the 2 possessed by Bill are "an increase over and above [the number of shells that Betsy has]." During the process of text integration, SELAH first decides that the phrase "Bill has 2 more shells ..." means that Bill does indeed have a set of two shells:

```plaintext
&;&;&;&;&;&;&;&;&;&;& & SENTENCE #1 &;&;&;&;&;&;&;&;&;&;&
-------------------------------
MAKE SET (2): BILL 2
tense = PRESENT  time = 2
-------------------------------
&;&;&;&;&;&;&;&;&;& & END SENTENCE #1 &;&;&;&;&;&;&;&;&
```

The specific referent, "than [the number of shells that Betsy has]." prompts SELAH to search memory for a set of shells owned by Betsy:

```plaintext
&;&;&;&;&;&;&;&;&;&;& & SENTENCE #2 &;&;&;&;&;&;&;&;&;&;&
Pronominal Reference: BETSY
+++++++++++++++++++++++
TEXT INTEGRATION REQUEST
"more than" BETSY has
+++++++++++++++++++++++
(Searching memory ...)
REFERENCE(BETSY) HAS A PREVIOUS SET

"BILL has 2 in-addition-to BETSY's 3"
```

Following the integration of the first two sentences, SELAH interprets the second sentence to mean "Bill has 2 shells in addition to [the 3 shells that Betsy has]." At this point, since "Bill has 2 in addition to those Betsy has," SELAH infers that Bill also has a number of shells equal to the number possessed by Betsy, that is, SELAH infers that Bill also has 3 other shells:

¹⁶ Relational expressions such as more than require EDUCE to handle ellipsis, that is the omission of a number of words from a phrase which are expected to be understood. For example, EDUCE interprets the phrase "Bill has 2 more shells than Betsy" as "Bill has 2 more shells than Betsy [has shells]." The words in brackets, [has shells], represent the ellipsis.
SELAH also makes two other inferences. First, since Bill has 2 shells "in addition to his other 3," a 

*Join* situation is inferred. SELAH also infers that this *Join* will result in a combined set of shells also 

owned by Bill.

Of particular importance is that SELAH understands the relational *more than* expression without 

resorting to a "large set, small set, difference set" schema. The model's assumption is that young chil-

dren have a solid understanding of the non-relational use of the word "more." As shown in the previous 

example, SELAH relies on this non-relational understanding even in relationally worded (*more than*) 

situations. The *in addition to* interpretation for "more" implies a *join* situation and provides a pro-

cedural interpretation from a completely static natural language description. Of course, even with the 

procedural re-interpretation, SELAH exposes some of tacit text integration processes which make rela-

tional problems the most difficult problems for children to solve. More specifically, SELAH had to make 

three inferences, a high number relative to the entire set of problems that SELAH solves. Table 6 sum-

marizes some of SELAH's processing on this *more than* relational problem.
Table 6.
A summary of SELAH's processing on a relational "more than" problem

| Betsy has 3 shells. Bill has 2 more shells than Betsy. How many shells does Bill have? | Sentence |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | Average |
| **Processing Summary** | make set | make set (Bill 2); referent found; infer JOIN; infer Bill also has an amount = to Betsy's 3; infer Bill's total is result of action | Bill ? |
| **Concepts in Memory** | [Betsy's 3] | [JOIN Bill's 2] | [JOIN Bill's 2] |
| | | [TO (Bill's 3)] | [TO (Bill's 3)] |
| | | [RE Bill's total] | [RE Bill's 2] |
| # of concepts | 1 | 3 | 3 | 2.33 |
| # of inferences | 0 | 3 | 0 | 1.00 |

3.6.4.2. "Fewer Than"

Unlike the word "more," the word "fewer" is understood by EDUCE in a relational fashion only. In the context of expectation based parsing, EDUCE always expects to encounter the word "than" after encountering the word "fewer." The reason is that in isolation the word "fewer" is a negative term (cf. Clark, 1969), i.e., I cannot have a negative amount representing the difference. The implication is that the relational expression "fewer than" is always understood by the model in a comparative sense. For example, consider the problem:

Betsy has 5 shells. Bill has 2 fewer shells than Betsy. How many shells does Bill have?

EDUCE interprets the second sentence to mean that "Bill has [an unknown number of shells] which is 2 fewer than [the number of shells that Betsy has]." An important difference in EDUCE's representation of this sentence and the more than example above is that Bill is not in possession of the 2 shells that are
described in the sentence. That is, the second sentence describes a set of shells that Bill has, but it does not explicitly indicate any part of that amount here:

&&&&&&&&&&&&& SENTENCE #2 &&&&&&&&&&&&&

-----------------------------------------------
MAKE SET (2): BILL ?
tense = PRESENT  time = 2
-----------------------------------------------

The specific referent, "than [the number of shells that Betsy has]," prompts SELAH to search memory for a set of shells owned by Betsy:

Pronominal Reference: BETSY
+++++++++++++++++++++++++++++++++++++++++++++++++++++
TEXT INTEGRATION REQUEST
"fewer than" BETSY has
+++++++++++++++++++++++++++++++++++++++++++++++++++++
(Searching memory ...)

REFERENCE(BETSY) HAS A PREVIOUS SET

"BILL has [an unknown number] which
is 2 fewer than BETSY's 5"

Following the integration of the two sentences, SELAH interprets the second sentence to mean "Bill has an unknown number of shells and those are 2 fewer than the 5 that Betsy has." At this point, SELAH interprets the fewer than description to imply a "take away" situation. Thus, SELAH (i) infers a Separate-From action and (ii) infers the needed transformation from "2 fewer than 5" to the procedural interpretation "5 separate 2." SELAH concludes its processing by inferring that Bill's unknown number is now described as being the result of the action of separating 2 from Betsy's 5:

******************************************************************************
!!! INFERRED !!!
Procedural SEPARATE_FROM situation
******************************************************************************
INSTANTIATE ACTION SEPARATE-FROM
******************************************************************************

17 The transformation from "2 fewer than 5" to "5 separate 2" is currently counted as an inference. As pointed out by Spanos, Rhodes, Dale and Crandall (1988), the phrase "x less than y" is syntactically confusing. Students tend to translate the expression in a left to right fashion, for example, they often write (x-y) rather than the mathematically correct (y-x).
The third and last sentence of the problem focuses attention on the unknown number in Bill's set (2).

Table 7 summarizes SELAH's processing on this fewer than relational problem.

Two points are especially important in this example. First, EDUCE must initially represent the expression "fewer than" in a relational manner. Unlike the situation where "Bill has 2 more than"

<table>
<thead>
<tr>
<th>Betsy has 5 shells, Bill has 2 shells fewer than Betsy. How many shells does Bill have?</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Processing Summary</td>
<td>make set</td>
</tr>
<tr>
<td>Concepts in Memory</td>
<td>[Betsy's 5]</td>
</tr>
<tr>
<td></td>
<td>[FR Bil's?]</td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
</tr>
</tbody>
</table>
results in the interpretation: "Bill (does indeed) have 2 plus some more." Bill does not have the 2 in the expression: "Bill has 2 fewer than." The second and related point is that the difference between two sets is not represented as a unique set in "fewer than" problems. Rather, SELAH maps this difference relation onto a procedural interpretation where the difference relation is interpreted in the sense of an operator rather than the cardinality of some set. Appendix B provides annotated scripts for other relational problems that SELAH solves.

3.7. Implications For a Computational Reading Model

The design and development of this model of mathematical reading continues to provide new levels of detail toward understanding why some problems are more difficult to solve than others. While two problems may have the same semantic structure and the same unknown and involve identical arithmetic computations, the "translation" steps of the reading model highlight important tacit problem solving processes which potentially affect problem difficulty; for example, the inclusion of a word or phrase can increase the connectivity of the text and thereby facilitate an understanding of the mathematical relationship between two sets. The EDUCE and SELAH components constitute an attempt to further define that network to be sensitive to the juncture between reading comprehension and mathematical comprehension.

More specifically, the EDUCE and SELAH components focus attention on tacit processes involved in the extraction of mathematical relationships from linguistically stated word problems. As shown in the previous examples, both components monitor cognitive processes during problem solution, for example the relative loads on working memory during sentence-level reading and text integration. Of specific interest is whether the cognitive measures from the model correlate with children's success on those same problems. The next chapter explores this possibility.

---

18 In fact, Riley and Greeno (1988) propose this extension to their model, that is, treating the values in difference relations as values to operators since the predictions of the Riley and Greeno (1988) model do not match children's success on relational problems. One cited reason for the model's inconsistent predictions was that the difference between two sets should not be represented as a set in a part-whole schema like the sets in the union between two sets.

19 A summary of the implementation details of the two components is given in Appendix D.
CHAPTER 4

Experiments

This chapter presents the results of three experiments which correlate children’s probability of solution on a benchmark set of problems with the monitored performance of the computer model’s two components, EDUCE and SELAH, on those same set of problems. Experiment 1 isolates measures of text integration in the SELAH component which account for a significant proportion of the variance in children’s solution probability. Experiment 2 isolates EDUCE’s sentence level reading processes which account for a significant proportion of the variance in children’s solution probability. Experiment 3 combines the significant sentence-level and text integration variables to establish a final set of predictor equations for solution probability.

4.1. Experiment 1

4.1.1. Method

In the present experiment, children’s probability of solution was analyzed in exploratory regression analyses as a function of 4 predictor variables, where the predictor variables were measures of text comprehension processes as simulated by the text integration component, SELAH.

4.1.1.1. Materials

The word problems solved by the reading model are a classic benchmark set of 18 problems from the Riley and Greeno (1988) and Riley, Greeno, and Heller (1983) experiments. A sample problem is shown below in Figure 4. A complete set of the 18 different problem types (or problem frames) is given in Appendix A. The problems are elementary addition and subtraction word problems containing sentences which describe two known quantities and a third unknown quantity. The problems are one-step
Kathy had 8 soda cans. Then Jacob gave her 3 soda cans. How many soda cans does Kathy have now?

Figure 4.
A sample arithmetic word problem

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 numbers in a problem range from 2-10 with an answer always amounting to 10 or less.

The solution probabilities of the 18 problem types are the results of previous research with children (Riley et al., 1983, 1988). The participants in these studies were children from kindergarten, first, second, and third grades. In the empirical studies, problems were read to the children in two problem solving formats: both with and without the presence of manipulatives (e.g., wooden blocks on a table). In one case, children were allowed to use manipulatives to aid in their solutions, where in the other case children were asked to solve the problems without manipulatives. Probabilities of solution were thus generated for each of the 18 problem types for each of the four grade levels both with and without the use of blocks. Table 8 and Table 9 summarize the probability of solution results when children did and did not use blocks, respectively.

4.1.1.2. Procedure

The 18 problems were input to the reading model one at a time. For each problem, the model read and solved the problem while printing out a record of its text integration processes. The output for each problem was then analyzed to obtain scores for the variables described in the next section. Detailed examples of SELAH's output are shown in Appendix B.
Table 8.
Proportions of correct solutions (with blocks)

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Grade K</th>
<th>Grade 1</th>
<th>Grade 2</th>
<th>Grade 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine 1</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.26</td>
<td>0.50</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>0.43</td>
<td>0.56</td>
<td>0.90</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>0.22</td>
<td>0.33</td>
<td>0.70</td>
<td>1.00</td>
</tr>
<tr>
<td>6</td>
<td>0.13</td>
<td>0.22</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>Change 1</td>
<td>0.87</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.61</td>
<td>0.56</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>0.91</td>
<td>0.78</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>0.09</td>
<td>0.28</td>
<td>0.80</td>
<td>0.95</td>
</tr>
<tr>
<td>6</td>
<td>0.22</td>
<td>0.39</td>
<td>0.70</td>
<td>0.80</td>
</tr>
<tr>
<td>Compare 1</td>
<td>0.17</td>
<td>0.28</td>
<td>0.85</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>0.04</td>
<td>0.22</td>
<td>0.75</td>
<td>0.95</td>
</tr>
<tr>
<td>3</td>
<td>0.13</td>
<td>0.17</td>
<td>0.80</td>
<td>1.00</td>
</tr>
<tr>
<td>4</td>
<td>0.17</td>
<td>0.28</td>
<td>0.90</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>0.17</td>
<td>0.11</td>
<td>0.65</td>
<td>0.75</td>
</tr>
<tr>
<td>6</td>
<td>0.00</td>
<td>0.06</td>
<td>0.35</td>
<td>0.75</td>
</tr>
</tbody>
</table>

From Riley and Greeno (1988)

4.1.1.3. Predictor Variables

The probability of solutions obtained in the Riley and Greeno studies were used in regression analyses as a function of 4 predictor variables. The variables represent computational measures of SELAH's processes 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, a simplified example is presented in Table 10. Summaries of text integration scores for each variable for all 18 problems are given in Appendix B. A summary of the 4 variables is given below:
Table 9.
Proportions of correct solutions (without blocks)

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Grade K</th>
<th>Grade 1</th>
<th>Grade 2</th>
<th>Grade 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine 1</td>
<td>0.74</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>0.70</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.04</td>
<td>0.39</td>
<td>0.85</td>
<td>0.85</td>
</tr>
<tr>
<td>4</td>
<td>0.13</td>
<td>0.39</td>
<td>0.70</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>0.22</td>
<td>0.33</td>
<td>0.55</td>
<td>0.75</td>
</tr>
<tr>
<td>6</td>
<td>0.17</td>
<td>0.44</td>
<td>0.55</td>
<td>0.85</td>
</tr>
<tr>
<td>Change 1</td>
<td>0.70</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>0.61</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.22</td>
<td>0.61</td>
<td>0.80</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>0.30</td>
<td>0.61</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>0.09</td>
<td>0.33</td>
<td>0.75</td>
<td>0.95</td>
</tr>
<tr>
<td>6</td>
<td>0.17</td>
<td>0.39</td>
<td>0.65</td>
<td>0.90</td>
</tr>
<tr>
<td>Compare 1</td>
<td>0.13</td>
<td>0.33</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>2</td>
<td>0.13</td>
<td>0.17</td>
<td>0.65</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>0.09</td>
<td>0.33</td>
<td>0.60</td>
<td>0.90</td>
</tr>
<tr>
<td>4</td>
<td>0.04</td>
<td>0.28</td>
<td>0.80</td>
<td>0.90</td>
</tr>
<tr>
<td>5</td>
<td>0.13</td>
<td>0.11</td>
<td>0.35</td>
<td>0.75</td>
</tr>
<tr>
<td>6</td>
<td>0.13</td>
<td>0.22</td>
<td>0.15</td>
<td>0.60</td>
</tr>
</tbody>
</table>

From Riley and Greeno (1988).

(1) **AVGMEM.** The average number of conceptual units in memory. This is the running sum of the number of conceptual units which appear in the model's working memory at the end of each sentence divided by the number of sentences in the problem. A conceptual unit is defined as either (i) a single conceptual set (e.g., [Jacob's 8]); (ii) an arithmetic action (e.g., JOIN or SEPARATE_FROM); or (iii) an action and conceptual set(s) that have been "chunked" together.

In Table 10, 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]"
are chunked into two concepts: "[SEPARATE Kathy's 3] [FROM Jacob's 8]." In Table 10, 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 \((1+2+3)/3 = 2.00\).

(2) **AVGINF.** The average number of inferences made. This is the running sum of the number of inferences which are made divided by the number of sentences in the problem. An inference is defined here as: (i) establishing a relationship between two sets when that relationship is not described in the text or (ii) instantiating an arithmetic action (e.g., *Join*) when the text does not mention a significant action. Text integration and arithmetic action inferences are made when the conceptual representation of a new sentence lacks the explicit information needed to establish text.

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>Concepts in Memory</th>
<th>number of concepts</th>
<th>number of inferences</th>
</tr>
</thead>
<tbody>
<tr>
<td>make set</td>
<td>1</td>
<td>[Jacob's 8]</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>INFER &quot;3 to Kathy&quot;</td>
<td>2</td>
<td>[SEP Kathy's 3]</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>are part-of 8;</td>
<td></td>
<td>[FR Jacob's 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>explicit SEP-FROM</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>explicit result</td>
<td>3</td>
<td>[SEP Kathy's 3]</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>of SEP-FROM</td>
<td></td>
<td>[FR Jacob's 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RE Jacob's ?]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Average</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table 10.**
Example of scoring for AVGMEM and AVGINF variables

---

1 As noted previously, two conceptual sets and an associated action are chunked only if both sets have known quantities. That is, [Kathy's zone] [SEPARATED_FROM] [Jacob's 8] remains three concepts. Performing such chunking when the quantities are known reflects the fact that all the information needed to solve the problem (i.e., quantity-operation-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.
connections between conceptual sets or instantiate an appropriate arithmetic action. In the example, an inference is made in the second sentence. Here, the model infers that the 3 cans that Jacob gave to Kathy are "part of his previously established set of 8." (An example sentence which would not have required this inference is: "Jacob gave 3 of his cans to Kathy." Here the phrase "of his cans" generates a direct link to the previous "set of 8." so a text integration inference would not be required). In the example of Table 11, the inference that "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 since 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 \((0+1+0)/3 = 0.33\)

(3) **MEM+INF.** 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 reflects the theory that children's total processing capacity is made up of (i) what they must remember as well as (ii) what must be devoted to executing basic operations, such as making inferences (Case, 1982). 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.

(4) **REACT.** This variable indicates whether a previously instantiated arithmetic action must be suppressed and changed to a new action (either from JOIN to SEPARATE_FROM or vice versa). Unlike the three continuous variables above, this variable is a dichotomous variable. A positive value of this variable indicates that the reading model prematurely instantiated an arithmetic action based on a keyword strategy (e.g., "altogether" means JOIN; "less" means SEPARATE_FROM) but had to suppress this erroneous assumption based on other processing.
The final scores for the four text integration variables for all 18 problems are shown in Table 11.

4.1.2. Results

To investigate the effects of the variables, four predictors were established to be used in a multiple linear regression analysis. These variables are described in the Method section and summarized in Table 12.

The four predictor variables were entered into a regression equation with the probability of solutions for a specific grade level as the dependent variable. For each grade level, the analysis proceeded in two stages. First, a stepwise regression was conducted with a liberal acceptance criterion: if a variable's

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>AVGMEM</th>
<th>REACT</th>
<th>AVGINF</th>
<th>MEM+INF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine 1</td>
<td>2.00</td>
<td>0</td>
<td>0.33</td>
<td>2.33</td>
</tr>
<tr>
<td>2</td>
<td>2.25</td>
<td>0</td>
<td>0.25</td>
<td>2.50</td>
</tr>
<tr>
<td>3</td>
<td>2.75</td>
<td>0</td>
<td>0.75</td>
<td>3.50</td>
</tr>
<tr>
<td>4</td>
<td>2.75</td>
<td>0</td>
<td>0.75</td>
<td>3.50</td>
</tr>
<tr>
<td>5</td>
<td>2.33</td>
<td>1</td>
<td>1.67</td>
<td>4.00</td>
</tr>
<tr>
<td>6</td>
<td>3.25</td>
<td>1</td>
<td>1.25</td>
<td>4.50</td>
</tr>
<tr>
<td>Change 1</td>
<td>2.00</td>
<td>0</td>
<td>0.00</td>
<td>2.00</td>
</tr>
<tr>
<td>2</td>
<td>2.00</td>
<td>0</td>
<td>0.33</td>
<td>2.33</td>
</tr>
<tr>
<td>3</td>
<td>3.00</td>
<td>0</td>
<td>0.00</td>
<td>3.00</td>
</tr>
<tr>
<td>4</td>
<td>3.00</td>
<td>0</td>
<td>0.25</td>
<td>3.25</td>
</tr>
<tr>
<td>5</td>
<td>3.00</td>
<td>0</td>
<td>0.00</td>
<td>3.00</td>
</tr>
<tr>
<td>6</td>
<td>3.00</td>
<td>0</td>
<td>0.25</td>
<td>3.25</td>
</tr>
<tr>
<td>Compare 1</td>
<td>2.33</td>
<td>0</td>
<td>1.00</td>
<td>3.33</td>
</tr>
<tr>
<td>2</td>
<td>2.33</td>
<td>0</td>
<td>1.00</td>
<td>3.33</td>
</tr>
<tr>
<td>3</td>
<td>2.33</td>
<td>0</td>
<td>1.00</td>
<td>3.33</td>
</tr>
<tr>
<td>4</td>
<td>2.33</td>
<td>0</td>
<td>1.00</td>
<td>3.33</td>
</tr>
<tr>
<td>5</td>
<td>3.00</td>
<td>0</td>
<td>1.33</td>
<td>4.33</td>
</tr>
<tr>
<td>6</td>
<td>3.00</td>
<td>1</td>
<td>1.33</td>
<td>4.33</td>
</tr>
</tbody>
</table>
contribution was significant at even the .15 level, the variable was included in the second stage of the analysis. The second stage was a regression analysis using those variables which met the .15 criterion in the first stage.

Table 13 summarizes the results of the four regression analyses when the dependent variable was solution probability for children in kindergarten through third grade when they did not have blocks available. For each grade, the variables listed in Column 2 are those that met the .15 criterion in the stepwise
regression: they appear in the order that they were selected. Column 3 presents the proportion of total variance accounted for by the subset of variables in that row and all preceding rows for each grade. (The double line indicates a new grade-level and thus an independent regression analysis). For example, in grade K, MEM+INF accounts for .537 of the variance and MEM+INF and REACT together account for .690 of the variance. Columns 4 and 5 present the regression coefficients and standard errors for the selected variables. The last column presents the p values (two-tailed) associated with each of the regression coefficients. For example, in grade K, MEM+INF was significant at the .000 level and REACT was significant at the .016 level.

Table 14 summarizes the results of four other regression analyses when the dependent variable was solution probability for children in kindergarten through third grade when they did have blocks available.

4.1.3. Discussion

An important result from this data is that MEM+INF accounts for a significant amount of variance in all of the eight analyses: MEM+INF is significant across all the four grade levels (K-3) when children do not have blocks available and for all four grade levels (K-3) when they do use blocks. Interestingly, AVGMEM (the average number of concepts that must be held in working memory) and AVGINF (the

<table>
<thead>
<tr>
<th>Grade</th>
<th>Variable</th>
<th>Variance Accounted For (R²)</th>
<th>Regression Coefficient</th>
<th>Standard Error</th>
<th>p Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>MEM+INF</td>
<td>.560</td>
<td>-.40</td>
<td>.09</td>
<td>.000</td>
</tr>
<tr>
<td>1</td>
<td>MEM+INF</td>
<td>.679</td>
<td>-.39</td>
<td>.07</td>
<td>.000</td>
</tr>
<tr>
<td>2</td>
<td>MEM+INF</td>
<td>.522</td>
<td>-.18</td>
<td>.04</td>
<td>.001</td>
</tr>
<tr>
<td>3</td>
<td>MEM+INF</td>
<td>.236</td>
<td>-.06</td>
<td>.03</td>
<td>.041</td>
</tr>
</tbody>
</table>
average number of inferences that must be made in order to establish an integrated text) do not account for significant amounts of the variance when considered individually. However, the combination of "concepts to remember" and "inferences made" (MEM+INF) serves as a consistent and highly significant predictor across grade levels. One explanation is that a slight increase in either the number of concepts that must be remembered or the number of inferences required is not enough, in and of itself, to cause a working memory overload.

In addition, it is noteworthy that REACT appears as a predictor only when children are not able to physically represent some of the sets with blocks. Interestingly, the sign of the REACT regression coefficient changes from positive for the youngest children (kindergarten and first grade) to negative for the oldest children (third grade). The presence of REACT (that is, having to suppress a previously instantiated arithmetic action in favor of another) apparently causes difficulty for the oldest students. One explanation is that very young children have not yet started to rely on the erroneous keyword strategies, such as "altogether means add(+)" whereas older children are more likely to have experienced some degree of success with strategies that encourage a scanning of the text for numbers and surface operation cues.

The decision to include REACT as a predictor variable was based on previous research results which show that students often make wrong operation errors in line with an initial (although erroneous) activation of an arithmetic action (Cummins et al., 1988; DeCorte and Verschaffel, 1985). For example, in the word problem shown in Figure 5, the most likely incorrect answer is eleven (11), that is, students will add the two numbers in the problem rather than subtract 3 from 8. According to the model, the presence of the keyword "altogether" initially causes young problem solvers to instantiate the JOIN arithmetic action. If they are not able to suppress this initial activation, a resulting representation of the problem involves "3," "8" and "JOIN," leading to an incorrect choice of addition and an answer of "11."

The notorious use of keyword strategies is verified in these results with the oldest problem solvers when
Jacob and Kathy have 8 soda cans altogether. Jacob has 3 soda cans. How many soda cans does Kathy have?

Figure 5.
A problem involving a potential change from JOIN to SEPARATE_FROM

they do not use blocks ($p = 0.66$).

Before further discussion of these results, Experiment 2 explores the possibility that reading processes at the sentence level might also account for some of the variance in solution probability. By monitoring the amount of work it takes the parsing component to read the individual sentences in each problem, the analysis can address the possibility that children’s difficulty with arithmetic word problems is due in part to their inability to read and conceptualize the individual sentences in the problem prior to text integration processes.

4.2. Experiment 2

Experiment 2 considers the amount of work it takes the parsing component to read individual sentences in each problem. The analysis addresses the possibility that children’s difficulty with arithmetic word problems is due in part to their inability to read and conceptualize the individual sentences in the problem prior to text integration processes.

4.2.1. Method

In this experiment, children’s probability of solution was analyzed in regression analyses as a function of 3 predictor variables, where the predictor variables were sentence-level measures of the “amount of work” required to parse and establish conceptual representations of the meaning of individual sentences in the problem.
4.2.1.1. Materials and Procedure

The materials are the same 18 benchmark problems as used in Experiment 1. The sentences in each of the 18 problems were input to the parsing model one at a time. For each sentence in each problem, the parsing model read and built a conceptual representation of that sentence. 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.

4.2.1.2. Predictor Variables

The probability of solutions obtained in the Riley and Greeno (1983, 1988) studies were used in regression analyses as a function of 3 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 below:

1. AVGPOST. The average number of expectations that are posted in the sentences of a problem. As described in Chapter 3, 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 an expectation is not the same as satisfying an expectation (see AVGSAT). Some expectations may be posted but never satisfied.

2. AVGSAT. 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 sum of the number of times the parser checks all the outstanding requests which are not yet satisfied. In
general, for each sentence, the parser checks all previously posted 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.

(3) **AVGSTM.** The average number of concepts in short-term memory (STM) while reading sentences in a problem. 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 one conceptual representation as expectations become satisfied and “chunk” these concepts together (e.g., [Kathy <POSSESSES> [soda can object]]. The parser monitors the total number of concepts that existed in 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.

The final scores for the three sentence-level variables for all 18 problems are shown in Table 15.

4.2.2. **Results**

To investigate the effects of the variables, three predictors were established to be used in a multiple linear regression analysis. These variables are described in the Method section and summarized in Table 16.

The three predictor variables were entered into a regression equation with the probability of solutions for a specific grade level as the dependent variable. For each grade level, the analysis proceeded in two stages as in Experiment 1.

Table 17 summarizes the results of the four analyses (one for each grade) when the dependent variable was solution probability for children in kindergarten through third grade when they did not have blocks available. Table 18 summarizes the results of four other analyses when the dependent variable was solution probability for children in kindergarten through third grade when they did have blocks avail-
Table 15. 
Actual scores of the predictor variables used in Experiment 2

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>AVGPOST</th>
<th>AVGSAT</th>
<th>AVGSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Combine 1</td>
<td>2.65</td>
<td>0.63</td>
<td>3.18</td>
</tr>
<tr>
<td>2</td>
<td>2.54</td>
<td>0.65</td>
<td>3.04</td>
</tr>
<tr>
<td>3</td>
<td>2.38</td>
<td>0.65</td>
<td>2.76</td>
</tr>
<tr>
<td>4</td>
<td>2.38</td>
<td>0.65</td>
<td>2.76</td>
</tr>
<tr>
<td>5</td>
<td>2.24</td>
<td>0.67</td>
<td>2.76</td>
</tr>
<tr>
<td>6</td>
<td>2.29</td>
<td>0.65</td>
<td>2.76</td>
</tr>
<tr>
<td>Change 1</td>
<td>2.53</td>
<td>0.68</td>
<td>2.76</td>
</tr>
<tr>
<td>2</td>
<td>2.78</td>
<td>0.75</td>
<td>2.83</td>
</tr>
<tr>
<td>3</td>
<td>2.77</td>
<td>0.65</td>
<td>2.86</td>
</tr>
<tr>
<td>4</td>
<td>3.12</td>
<td>0.75</td>
<td>2.96</td>
</tr>
<tr>
<td>5</td>
<td>2.62</td>
<td>0.68</td>
<td>2.92</td>
</tr>
<tr>
<td>6</td>
<td>2.80</td>
<td>0.72</td>
<td>2.96</td>
</tr>
<tr>
<td>Compare 1</td>
<td>2.82</td>
<td>0.60</td>
<td>2.94</td>
</tr>
<tr>
<td>2</td>
<td>2.82</td>
<td>0.60</td>
<td>2.94</td>
</tr>
<tr>
<td>3</td>
<td>2.59</td>
<td>0.63</td>
<td>2.88</td>
</tr>
<tr>
<td>4</td>
<td>2.59</td>
<td>0.63</td>
<td>2.88</td>
</tr>
<tr>
<td>5</td>
<td>2.59</td>
<td>0.63</td>
<td>2.88</td>
</tr>
<tr>
<td>6</td>
<td>2.59</td>
<td>0.63</td>
<td>2.88</td>
</tr>
</tbody>
</table>

Table 16. 
Summary of the predictor variables used in Experiment 2

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVGPOST</td>
<td>average number of expectations posted</td>
</tr>
<tr>
<td>AVGSAT</td>
<td>average number of expectations satisfied</td>
</tr>
<tr>
<td>AVGSTM</td>
<td>average number of concepts in STM</td>
</tr>
</tbody>
</table>
Table 17.
Results of regression analysis without blocks for Experiment 2

<table>
<thead>
<tr>
<th>Grade</th>
<th>Variable</th>
<th>Variance Accounted For (R²)</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>

4.2.3. Discussion

The main result from these analyses is that sentence-level variables account for significant amounts of variance (p < .05) only at the kindergarten and first grade levels when children use blocks. In the analyses using probabilities when children use blocks, AVGSAT and AVGSTM account for 39% of the variance in grade K and AVGSAT and AVGSTM account for 34% of the variance in grade 1. No variable
appears at the 2nd or 3rd grade levels even with the liberal probability to enter criterion of $p=.15$. The
fact that older children's solution probabilities do not vary with the "amount of conceptual work" at the
sentence level is not surprising. By the second and third grade, the majority of children are reading and
interpreting the meaning of most sentences with relatively little effort. These results confirm eye move-
ment results from DeCorte et al. (1990) and Hegarty et al. (1992) which show that the time for an initial
reading of the problem (the translation phase) does not significantly vary between problems that differ in
probability of solution.\footnote{The eye movements results are from college-level students and not elementary school children. Hegarty et al.'s (1992) results also only address Compare problems.} In agreement with Experiment 2, Hegarty's eye movement results show that nei-
ther high accuracy nor low accuracy college 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. That is, difficult
problems require additional text integration processing, a result consistent with the findings of Experiment
1.

In the analyses using probabilities without blocks, AVGSTM and AVGSAT account for 27% of the
variance in grade K, although neither variable is significant ($p<.05$). No other variable is significant at
the 1st, 2nd or 3rd grade levels ($p < .05$).

In short, the results of Experiment 2 extend previous results (Hegarty et al., 1992; Lewis & Mayer,
1987) to a new dependent measure, namely, the amount of parsing effort rather than the amount of time
to read or error rate. That is, children's solution probabilities do not vary with the "amount of work"
required to read the sentences in a problem. However, it is possible that EDUCE is not yet monitoring
the "critical" processes at the sentence level. For example, it became evident during the design of
EDUCE that relational expressions contain an ellipsis that does not occur in any of the other word prob-
lems. For example, the sentence below shows a typical relational sentence with the ellipsed portion in
brackets.
Kathy has 5 more soda cans than Jacob [has soda cans].

While EDUCE is able to handle this ellipsis, the monitoring is not fully sensitive to the amount of processing that such constructions require. An ability to monitor additional tacit reading tasks such as ellipsis remains a goal of future work.

4.3. Experiment 3

In order to unify the results from the previous two experiments, Experiment 3 combines the text integration variables that were entered in the stepwise regression in Experiment 1 with those variables that were entered in the stepwise regression in Experiment 2. This combination of sentence-level and text integration variables should reveal those variables which are most sensitive to the changes in children's solution probability. The results of Experiment 3 also determine a set of equations to predict a given problem's probability of solution.

4.3.1. Method

In this experiment, children's probability of solution was analyzed in regression analyses as a function of both the text integration variables from Experiment 1 and the sentence-level variables from Experiment 2 that were significant at the \( p < .05 \) level. The materials and procedure are the same as those of Experiment 1 and 2. A summary of the text integration and sentence-level predictor variables are described in Experiments 1 and 2, respectively.

4.3.2. Results

Table 19 summarizes the results of the analyses when the dependent variable was solution probability for children in kindergarten through third grade when they did not have blocks available.

Table 20 summarizes the results of the analysis when the dependent variable was solution probability for children in kindergarten through third grade when they did have blocks available.
Table 19.
Results of Regression Analysis without Blocks in Experiment 3

<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>MEM+INF REACT</td>
<td>.537</td>
<td>-0.37</td>
<td>0.07</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.690</td>
<td>0.32</td>
<td>0.12</td>
<td>.016</td>
</tr>
<tr>
<td>1</td>
<td>MEM+INF REACT</td>
<td>.646</td>
<td>-0.47</td>
<td>0.07</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.764</td>
<td>0.36</td>
<td>0.13</td>
<td>.015</td>
</tr>
<tr>
<td>2</td>
<td>MEM+INF</td>
<td>.721</td>
<td>-0.29</td>
<td>0.05</td>
<td>.000</td>
</tr>
<tr>
<td>3</td>
<td>MEM+INF REACT</td>
<td>.577</td>
<td>-0.08</td>
<td>0.03</td>
<td>.019</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.664</td>
<td>-0.12</td>
<td>0.06</td>
<td>.066</td>
</tr>
</tbody>
</table>

Table 20.
Results of Regression Analysis with blocks in Experiment 3

<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>MEM+INF AVGSAT</td>
<td>.560</td>
<td>-0.35</td>
<td>0.09</td>
<td>.001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.635</td>
<td>2.44</td>
<td>1.40</td>
<td>.101</td>
</tr>
<tr>
<td>1</td>
<td>MEM+INF AVGSAT</td>
<td>.679</td>
<td>-0.35</td>
<td>0.07</td>
<td>.000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.735</td>
<td>1.89</td>
<td>1.06</td>
<td>.095</td>
</tr>
<tr>
<td>2</td>
<td>MEM+INF</td>
<td>.522</td>
<td>-0.18</td>
<td>0.04</td>
<td>.001</td>
</tr>
<tr>
<td>3</td>
<td>MEM+INF</td>
<td>.236</td>
<td>-0.06</td>
<td>0.03</td>
<td>.041</td>
</tr>
</tbody>
</table>

4.3.3. Discussion

As expected, the MEM+INF text integration variable remains the dominant predictor across all grade levels, both when children did and did not use blocks. In fact, only one sentence-level variable appears (AVGSAT) in the case when children did use blocks (kindergarten and first grade), although this variable does not approach significance (p_k=.101 and p_f=.095). Clearly, the work involved during the integration stage of solution (e.g., inferring an (unmentioned) relationship between two sets) is more
critical for children than the work required to conceptualize the gist of natural language sentences.

4.4. General Discussion

Three significant results emerge from this study. First, a combined measure of the number of concepts to remember and the number of inferences to make while remembering those concepts is a consistent predictor of children's problem solving success. If one considers a child's total processing space as fixed (as suggested by Case, 1982), 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. In the context of arithmetic word problems, the inferences necessary for text integration are critical. An ability to arrive at a correct arithmetic operator is directly dependent on establishing the relationships 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 which 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; DeCorte & Verschaffel, 1985; and others).

The second result provides new hypotheses concerning children's persistent reliance on keyword strategies. While it is widely known that children often rely on surface clues for problem solving hints, the reading model offers a new explanation as to the source of the keyword difficulty. Many of the classic keywords associated with word problems imply arithmetic actions (e.g., JOIN), but, they do not always imply a one-to-one mapping to an arithmetic operation. While 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. For example, consider the following problem:

Zachary and Kathy found 8 cans altogether. Zachary found 3 cans.
Kathy found some cans. How many cans did Kathy find?

In the first sentence, the word "altogether" does imply the arithmetic action JOIN, that is, the [set of 8] could be made by JOINing those Zachary found with those that Kathy found. A conceptualization of the entire problem might be:


In the computer simulation, altogether always instantiates the arithmetic action JOIN; however, in the problem above, text integration processes suppress JOIN in favor of SEPARATE_FROM since [Zachary's 3] are part-of the [set of 8] and can be formed by SEPARATING 3 FROM the 8. Children who are not able to suppress the previously instantiated JOIN, however, may be left with a representation of [3], [8] and JOIN. An incorrect answer of eleven (11) may not strictly mean that a child read "altogether" and decided to add the two numbers in the problem. An inability to (i) infer that [Zachary's-3] are part-of the [set of 8] and/or (ii) suppress the previously instantiated JOIN results in a representation dominated by JOIN and independent sets.

A final result concerns the dominance of the text integration variables as compared to the sentence-level variables. While the sentence-level variables account for small amounts of the variance in the youngest children's solution probabilities in Experiment 2, text integration variables are clearly the strongest predictors of solution success, as shown in Experiment 3. Intuitively, this is not surprising since most children are able to read and understand individual sentences. However, the small amounts of variance accounted for by the sentence-level variables may be more a function of the selection of variables than the fact that nothing of interest is occurring at this level of comprehension. For instance, both Combine and Compare problems do not include action verbs and describe only static situations. More specifically, the Compare problems involve the complex relational expressions "more than" and "less than." While it is not entirely clear how to best quantify some of the sentence-level reading processes, the reading model has exposed processes that specifically occur while parsing relational sentences, for
example, ellipsis. As shown in recent eye movement experiments with these problems (Hegarty, Mayer & Green, 1992; Verschaffel, DeCorte, & Pauwels, 1992), the model’s ability to focus at fine-grained levels of processing can reveal tacit factors which may contribute to children’s difficulties with relational language and sentence-level processing in general.

In summary, children’s difficulties with arithmetic word problems are 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 two-fold. First, the processes of text comprehension and mathematics 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 which imply or even explicitly state the relationships between sets is a critical step towards helping those students who can not yet make the necessary inferences, as recently shown by Davis-Dorsey, Ross & Morrison (1992) and others. Children are often expected to make complex inferences required by sparsely worded problems while they are just beginning to read. The second educational implication, related to the first, is that a fine-grained classification of word problems is emerging. The reading model 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. As presented in the next chapter, determining the “next best problem to try” is a function of many factors, including the processes of text integration.
CHAPTER 5

A Network of Problems

5.1. Introduction

This chapter presents a difficulty-differentiated network of problems that includes a multiple number of rewordings for each “traditional” problem wording. Centered around the results from the regression experiments in the previous chapter, predictor equations are used to confirm previous research on the effects of problem wording and to extend the traditional measures of why one problem is easier or harder than another. The educational objectives are to increase teacher awareness of the multiple sources of problem difficulty and to show how and hypothesize why slight changes in problem wording can affect children’s solution success. The results of the analysis in this chapter is an extension of the traditional classification of word problems in terms of a more fine-grained “network” of problems to help teachers determine a “next best” problem to present.

The chapter proceeds as follows. First, the traditional measures of arithmetic word problem difficulty are introduced. New task characteristics of problem solving difficulty are then introduced. Initially, the specific effect of each new task characteristic on solution probability is not discussed but is reserved for later sections. The final sections present examples which highlight the types of problems where individual task characteristics have the greatest influence on changes in solution probability. In each example, task characteristics are introduced, individually and/or in combination, to generate new wordings. Depending on the task characteristic introduced, each reworded version is then classified as easier or harder than a traditional wording as found in Riley et al.’s (1983, 1988) benchmark set of problems. Where possible, empirical studies are cited which confirm a relative order of difficulty between two problems. In addition, the predictor equations as derived from the regression analyses in Chapter 4 are
used to confirm the empirical results and/or predict the effect of each task characteristic on the probability of solution. The result is sets of four to six potential rewordings in a relative order of difficulty for each of the traditional wordings.

5.2. Task Characteristics of Problem Difficulty

A number of task characteristics are emerging which individually and in combination affect the solution probability of arithmetic word problems. These task characteristics include the traditional measures of semantic structure, location of the unknown and type of arithmetic action as well as new measures including the sequence of events (e.g., chronological vs. out of sequence), the relationship between co-actors in the problem (e.g., the co-actors are brothers vs. no relation), narrative focus (e.g., whether a problem is consistently stated from the perspective of the protagonist) and the presence of certain language which facilitates text integration (e.g., *more, of them*) or implies a direct modeling or counting strategy.¹

5.2.1. Semantic Structure and Location of the Unknown

The semantic structure of a word problem and the location of the unknown serve as two primary measures of problem solving success, as confirmed by a number of longitudinal studies (Carpenter & Moser, 1984; Nesher & Katriel, 1977, Riley et al., 1983 and others). Changes in semantic structure alter the semantic relations between sets of objects as described in the problem text, such as *change of possession or location* (Change problems), *combinations* (Combine problems) and *comparisons* (Compare problems). When comparing these problem types to each other across the grade levels of K, 1, 2 and 3, the

¹ Three other task characteristics, the sequence of the given numbers, the inclusion of extraneous information and the use of real-world situations of interest and personalization, are not discussed here. Verschaffel and DeCorte (1990) found significant differences in solution strategies on problems starting with the smaller versus the larger given number. The use of varying amounts of extraneous information can clearly make a word problem more difficult (cf., Searle et al., 1974). General context effects such as the use of favorite object and the names of friends, etc. have been shown to help some students (Davis-Dorsey et al., 1992). It is assumed that affective factors such as personalization can facilitate children's interest, in line with the work of McLeod (1988). While these characteristics are considered important enough to be included in a final set of variables affecting problem difficulty, their open-ended nature is beyond the scope of the fine-grained differentiation presented here.
average Change problem ($p=0.71$) is in general easier than the average Combine problem ($p=0.65$) which is in turn easier than the average Compare problem ($p=0.43$), as shown in the Riley and Greeno (1988) data in Table 21.\footnote{These results are for the most part consistent with other studies. See Nesher, Greeno & Riley (1982) for further validation of the consistency of solution probabilities between varying semantic structures as measured in different empirical studies.} The statement that Change is easier than Combine which is easier than Compare must be qualified by the fact that changes in the location of the unknown in these types of problems significantly alters the solution probability (Hiebert, 1982). For example, Change problems which have an

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textbf{Problem Type} & \textbf{Grade K} & \textbf{Grade 1} & \textbf{Grade 2} & \textbf{Grade 3} & \textbf{Average} \\
\hline
Combine 1 & 0.74 & 1.00 & 1.00 & 1.00 & 0.94 \\
2 & 0.70 & 1.00 & 1.00 & 1.00 & 0.92 \\
3 & 0.04 & 0.39 & 0.85 & 0.85 & 0.53 \\
4 & 0.13 & 0.39 & 0.70 & 1.00 & 0.56 \\
5 & 0.22 & 0.33 & 0.55 & 0.75 & 0.46 \\
6 & 0.17 & 0.44 & 0.55 & 0.85 & 0.50 \\
All combine & 0.33 & 0.59 & 0.78 & 0.91 & 0.65 \\
\hline
Change 1 & 0.70 & 1.00 & 1.00 & 1.00 & 0.92 \\
2 & 0.61 & 1.00 & 1.00 & 1.00 & 0.90 \\
3 & 0.22 & 0.61 & 0.80 & 0.95 & 0.64 \\
4 & 0.30 & 0.61 & 1.00 & 1.00 & 0.73 \\
5 & 0.09 & 0.33 & 0.75 & 0.95 & 0.53 \\
6 & 0.17 & 0.39 & 0.65 & 0.90 & 0.53 \\
All change & 0.35 & 0.66 & 0.87 & 0.97 & 0.71 \\
\hline
Compare 1 & 0.13 & 0.33 & 0.65 & 1.00 & 0.53 \\
2 & 0.13 & 0.17 & 0.65 & 1.00 & 0.49 \\
3 & 0.09 & 0.33 & 0.60 & 0.90 & 0.48 \\
4 & 0.04 & 0.28 & 0.80 & 0.90 & 0.50 \\
5 & 0.13 & 0.11 & 0.35 & 0.75 & 0.34 \\
6 & 0.13 & 0.22 & 0.15 & 0.60 & 0.28 \\
All compare & 0.11 & 0.24 & 0.53 & 0.86 & 0.43 \\
\hline
All problems & 0.26 & 0.50 & 0.72 & 0.91 & 0.60 \\
\hline
\end{tabular}
\caption{Proportions of correct solutions for varying semantic structures}
\end{table}

From Riley and Greeno (1988).
initial set unknown (Change 5 & 6) are typically harder for most children than the superset unknown Combine problems (Combine 1 & 2). This of course can also vary across grade levels. For instance, by third grade, almost all children are solving the most difficult Change problems.

5.2.2. Action Type

Arithmetic word problem texts mention either explicitly or by implication one of two arithmetic actions: Join or Separate-From.\(^3\) Arithmetic actions are more specific to the situation as expressed in the problem text than an arithmetic operator that might be selected for an equation. Equations for arithmetic word problems involve the mathematical operators of addition or subtraction\(^4\) and deciding on an arithmetic operator is often considered to fall just short of arriving at the answer itself. In a transition from a natural language problem to a numerical solution, the mathematical equation is commonly referred to as the "bridge" connecting these representations (Polya, 1957). While equations provide one alternate representation of a situation originally expressed in natural language, experience has indicated that we do not want to teach children a "words to equation" mapping (NCTM, 1991). As simulated in the computer model, the solution of arithmetic word problems involves a number of intermediate representations in increasingly abstract forms and arriving at an arithmetic action, such as Join or Separating-From, is a critical link in the progression from words to equations. Arithmetic actions, however, do not imply arithmetic operators. For example, a word problem may be stated such that two subsets are Joined in order to form a third resulting set, resulting in a representation of the situation:

\[
\text{set}_1 \ JOINed \ to \ \text{set}_2 \ \text{resulting in} \ \text{set}_3
\]

If one of the subsets has an unknown quantity (e.g., set\(_2\) has some), a correct solution procedure cannot

\(^3\) It is possible for some wordings to also imply a direct modeling matching strategy, e.g., Hudson's (1983) "How many birds won't get a worm?" However, I am assuming a problem solving strategy as simulated in the computer model where all problems are solved by counting strategies as derived from either Join or Separate-From arithmetic actions.

\(^4\) Some classifications of arithmetic word problems also include the operators of multiplication and division. Since this work is focused on the very young reader, the assumption is that the word problems only involve the two easiest operators of addition and subtraction.
resort to adding (or joining) the two known quantities. Rather, the procedure must be altered to another arithmetic action, such as Add-On. As argued in Chapter 4, arriving at a representation with an arithmetic action is a necessary although not sufficient condition for arriving at the correct answer. In the example above, arriving at an arithmetic action does not directly imply the choice of an arithmetic operator. In this case, depending on the exact location of the unknown and the developmental level of the child, the problem solving process may continue by a number of strategies: (i) directly modeling the JOIN situation by adding on, (ii) proceeding to a more abstract representation of values and operator \(3 + \text{some} = 5\) and use a counting strategy such as count on from first \(3 \ldots 4\) (that's 1), 5 (that's 2)) or (iii) transforming the representation into a Separate-From situation and continue such as described in (i) or (ii) above.

In short, word problems of any semantic structure can be classified by the type of arithmetic action, either Join or Separate-From that is explicitly stated or implied in the problem text. This task characteristic of problem difficulty, however, focuses its explanations of problem difficulty almost exclusively on abstract semantic characteristics, that is, the differences in basic logico-mathematical competencies.

5.2.3. Situational Characteristics

The next three sections refer to task characteristics which integrate the search for logico-mathematical structure with an analysis of situational and linguistic structures. These "situational characteristics" address what Staub and Reusser (1992) call the presentational structure of word problems that describe "real-world" situations, specifically, situationally-rich Change problems. The presentational structure of a word problem includes:

A story's content, that is, the facts and events which constitute the "world" as referred to in a story text, such as relations among the events of the story, including temporal order, relations of causation, motivation, may be verbally described in many different ways (Staub and Reusser, 1992, p15).
5.2.3.1. Time Sequence

For problems which involve actions, e.g., the transfer of objects from one person to another, the correspondence between the order of events as expressed in the text and the order that they actually occurred is emerging as an important task characteristic, as argued by Reusser (1989) and verified by Staub and Reusser (1992). For example, Figure 6 shows a problem where the text presents a situation in the same temporal order of the actual occurrence of those events. The sentences in Figure 7, on the other hand, are not presented in their “natural” order; that is, the transaction from Sylvia to Dan today is described prior to the statement describing Sylvia’s amount yesterday.

5.2.3.2. Narrative Focus

In word problems that involve actions between people, the protagonist is the leading character in the “story problem” and the owner of the quantity of interest. Staub and Reusser (1992) define a consistent narrative focus as a presentational structure of a word problem that is constructed such that the

Sylvia had 5 marbles. Then Sylvia gave Dan 3 marbles. How many marbles does Sylvia have now?

Figure 6.
Chronologically presented time sequence

Today Sylvia gave Dan 3 marbles. Yesterday Sylvia had 5 marbles. How many marbles does Sylvia have now?

Figure 7.
Non-chronological time sequence
protagonist is fixed in the grammatical subject position. For example, if Steve is the protagonist of a
story, the sentence: "Steve got 2 floppy disks from Liz" is considered to have a consistent narrative
focus. The same sentence might be stated: "Liz gave 2 floppy disks to Steve" which creates an inconsistent narrative focus.

5.2.3.3. Related Co-Actors

In problems involving two people, the co-actors can be referred to by their true names or one co-
actor can be indirectly referred to in terms of his family relationship to the other co-actor. For example,
the two co-actors could be referred to in the problems as "David and Robert" or the co-actors could be
referred to as "David and his cousin." Unlike the first two presentational factors, time sequence and nar-
rative focus, which specifically address situationally-rich Change problems, the use of related co-actors is
applicable for any type of problem that involves two co-actors.

5.2.4. Facilitating Text Integration

The last task characteristic involves the use of specific "mathematical" words and phrases which
reinforce which set is a whole vs. a part. As fully discussed in Chapter 4 in the context of the computer
model, words and phrases such as more, of them, the rest and left cause a representation of the current
sentence to be "linked" to the representation of a previous sentence. As expressed in the language of van
Dijk and Kintsch's (1983) theory of text comprehension, the process of constructing an internal representa-
tion of the situation is facilitated by an already integrated text base, i.e., the propositional representation
of the textual input. On the other hand, problems which do not include such "mathematical" words
require the reader to infer the essential connections between sentences.

5.3. Towards a Network of Problems

Of interest in this chapter is how to organize the knowledge of problem types, student competencies
and different task characteristics in such a way as to facilitate the processes of assessment and tutoring.
As Carpenter et al. (1988) suggest, teachers would benefit from understanding the relation between semantic types, children’s solutions strategies and why one problem type is often more difficult than another. Yet clearly, it is not enough to know about the different types of semantic relations which occur in arithmetic word problems. When faced with students who are experiencing difficulties with a particular type of word problem, teachers need more than a suggestion to move the location of the unknown or to try a problem with a different semantic relation. While such suggestions may generate another problem which is indeed easier, the teacher is probably no closer to understanding what makes the first problem more difficult. While understanding that “this semantic relation with the unknown in this location is harder than that semantic relation with the unknown in that location” is a first step, the fact remains that a student still has difficulty with the harder problem. This is precisely the limitation of solely relying on the traditional measures of semantic structure and location of the unknown: keeping the semantic type and location of the unknown, there are no suggestions as to how this type of problem might be made easier for the student experiencing difficulty.

The next sections present examples of three types of problems and the associated task characteristics which affect problem difficulty in each type. The three types of problems include (i) significant action language, (ii) non-relational static language or (iii) relational static language. Of particular interest is the mapping between the types of problems and the individual task characteristic which have the greatest influence on solution probability. Each task characteristic is matched with the most applicable problem type, although some problem rewordings appear across all types.

5.3.1. Making Significant Action Language Problems Easier

Figure 8 shows Riley et al.’s Change 1 problem, a typical problem with significant action language. In Change 1 problems, a specific amount is explicitly joined to an already existing set with a known amount. The result of the join is unknown. While most children can easily solve this problem (70% success for grade K and 100% success for grades 1, 2 and 3), there are still at least three potential ways to
Alex had 5 marbles. Then Bethany gave Alex 3 marbles. How many marbles does Alex have now?

Figure 8.
Traditional Change 1

alter the traditional wording to make it easier: (i) use a consistent narrative focus; (ii) include a related co-actor and (iii) include "mathematical" words which increase text integration and reinforce strategies which perform arithmetic actions on isolated sets.

The first potential alternative to the traditional wording of Change 1 is to reword the second sentence such that the narrative focus is consistent, e.g., to put Alex in the grammatical subject position in all the sentences, as shown in Figure 9. In both versions, Alex is the protagonist of the story due to his mention in the first sentence. In the traditional wording, however, focus is shifted away from Alex when Bethany is introduced in the grammatical subject position in the second sentence. While Alex is the recipient of the 3 marbles in both versions, the consistent narrative focus more closely resembles the required arithmetic action of joining 3 marbles with Alex's original set of 5.

The second potential alternative to the traditional wording of Change 1 is to substitute a related co-actor (e.g., his sister) for the unrelated co-actor, Bethany, as shown in Figure 10. In the traditional word-

Alex had 5 marbles. Then Alex got 3 marbles from Bethany. How many marbles does Alex have now?

Figure 9.
Change 1 with consistent narrative focus
Alex had 5 marbles. Then his sister gave 3 marbles to Alex. How many marbles does Alex have now?

Figure 10.
Change 1 with related co-actor

ing of Change 1, Bethany is in no way related to Alex and therefore does not provide any information which might lead to a more integrated representation of the situation. By including a related co-actor, the protagonist, Alex, remains the “center” of focus during comprehension and the situation is more likely to be coordinated from his perspective.

The third potential alternative to the traditional wording of Change 1 is to include words which might facilitate the links between sentences by highlighting the relationship between quantities. As discussed in Chapter 3 in the context of the computer model, the word *more* generates an implicit reference to a previously mentioned set, i.e., “*more* than what?” In the case of Change 1, including the word *more* in the second sentence, e.g.,

Then Bethany gave Alex 3 *more* marbles.

identifies that the three given to Alex are related to but in addition to the five he already had in his possession. That is, *more* can help isolate the three that are given to Alex by highlighting that these should not be confused as being part of the five he already had.5

As shown in the work of Staub and Reusser (1992), certain problems with related co-actors are solved more quickly than problems with unrelated co-actors. Although no significant differences were found between problems with consistent and inconsistent narrative focus when the co-actors were related,

5 Of course, it is possible that including *more* will reinforce some childrens’ use of the keyword strategy “*more* means ADDITION.” If a child did have difficulty with the traditional wording but then correctly solved a version which included “*more*,” the teacher or learning environment would then have to determine whether the child was relying on this keyword strategy by presenting problems which include the word “*more*” but require SUBTRACTION.
consistent narrative focus problems were easier than inconsistent focus problems when the co-actors were unrelated.

Increasing solution probability on Change 1 problems with these suggestions is presented only as a working hypothesis for two reasons. First, specific Change 1 problems were not included in Staub's experiments, thus there is no direct evidence that changing to a consistent narrative focus or using related co-actors will help. Second, the predictor equations from the computer model are not sensitive to changes in narrative focus, co-actor relation or the inclusion of "more" in this particular problem. In general, however, Staub's results provide evidence that using related co-actors and a consistent narrative focus can facilitate comprehension on the most difficult problems involving significant action language.

5.3.2. Making Significant Action Language Problems Harder

The traditional Change 1 problem can be made potentially more difficult by (i) constructing a more sparsely worded problem or (ii) altering the presentational structure of the problems such that the text sequence does not match the natural order of events as they would happen in the real world. A complete network of problems must include these more difficult problems to challenge the advanced students and more important, to help isolate the rewording changes which cause a student to experience difficulty.

In all Change problems, including Change 1, specific language is used to indicate the relational time order of events (e.g., then, after that) or to refer to specific points in time (e.g., in the beginning, now). When relational time language is removed, children have to infer the order of events by relying on (i) the tense of sentences or (ii) a default strategy that sentences appear in chronological order. For example, the word "then" could be removed from the second sentence of the traditional Change 1 and "now" could be removed from the last sentence, as shown in Figure 11. In this problem, the first two sentences are in the past tense. A failure to utilize the default strategy that sentences appear in chronological order may cause children to infer that Bethany gave Alex three of his original five, i.e., "Alex had 5 and Bethany gave him 3 of them." Likewise, when the word "now" is removed, children must notice that
Alex had 5 marbles. (Then) Bethany gave Alex 3 marbles. How many marbles does Alex have (now)?

Figure 11.
Removing (time language) from Change 1

the question is posed in the present tense. Since the action occurred in the past, the desired quantity of marbles is that which Alex has now. There is no empirical evidence that the removal of these words would cause Change 1 to be more difficult. In addition, the model's monitoring is not currently sensitive to such changes. However, the model's parser, EDUCE, makes specific use of words like then and now, and their absence requires the use of a default comprehension strategy from the text integration component, SELAH. For example, if "then" is not included in the second sentence, SELAH defaults to a chronological ordering of events. In short, omission of the relational time words is hypothesized, although weakly, to cause Change 1 problems to be more difficult.

The second rewording change which is known to make Change 1 problems more difficult is to alter the time sequence of the problem. As shown in empirical studies by Rosenthal and Resnick (1974) and Staub and Reusser (1992), altering the time sequence such that the sentences are not in strict chronological order does indeed make the problem more difficult. As shown in Figure 12, such a change to the presentational structure causes the first sentence to refer to the transfer 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 this time-altered version is explained through the computer model when it solves these two problems like a novice who reads in a sentence-by-sentence fashion. In the first sentence of the traditional wording, Alex is understood to possess five marbles and the model creates a set of five for Alex. The second sentence is understood to be a transfer to Alex, thus the three that are transferred are JOINed to his previous five as described by the "transfer-
Today Bethany gave Alex 3 marbles. Yesterday Alex had 5 marbles. How many marbles does Alex have now?

Figure 12.
Altering the time sequence of Change 1

in situation in the first two sentences. The JOIN action is explicit in the situational representation and need not be inferred. A summary of the model’s processing for the traditional Change 1 problem is shown in Table 22.

The time-altered version highlights the model’s sensitivity to changes in problem wording, as discussed next. In the time-altered version, the transfer of three to Alex occurs in the first sentence. Since Alex does not previously have any in his possession, the model simulates the transfer as resulting in Alex’s set of 3 today. The second sentence indicates that Alex had 5 yesterday and the model makes

<table>
<thead>
<tr>
<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>transfer-in situation; explicit JOIN action</td>
<td>explicit result of JOIN</td>
<td></td>
</tr>
<tr>
<td>Concepts in Memory</td>
<td>[Alex’s 5]</td>
<td>[JOIN Beth’s 3] [TO Alex’s 5]</td>
<td>[JOIN Beth’s 3] [TO Alex’s 5] [RE Alex’s ?]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2.00</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
</tr>
</tbody>
</table>
another set for Alex with this amount. Since the model is reading in a sentence-by-sentence fashion it is not able to "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 model infers that a difference in time differentiates the two sets so it can JOIN Alex’s marbles-today and Alex’s marbles-yesterday. In short, when the computer model comprehends the time-altered problem, the transfer of 3 from Bethany to Alex does not instantiate a JOIN action. That action must be inferred at the end of the problem. A summary of the model’s processing for the time-altered version of Change 1 is shown in Table 23.

As discussed above, Staub and Reusser (1992) have shown that the time-altered version of Change 1 is significantly more difficult to solve than the traditional version. Because the computer model is also

<table>
<thead>
<tr>
<th>Table 23. Summary of model’s processing for time-altered Change 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Today Bethany gave Alex 3 marbles. Yesterday Alex had 5 marbles.</td>
</tr>
<tr>
<td>How many marbles does Alex have now?</td>
</tr>
<tr>
<td>Sentence</td>
</tr>
<tr>
<td>Processing Summary</td>
</tr>
<tr>
<td>Concepts in Memory</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>number of concepts</td>
</tr>
<tr>
<td>number of inferences</td>
</tr>
</tbody>
</table>

*The model solves this problem much like it solves a more static Combine problem. In the time-altered Change 1 problem, the action language does not facilitate the instantiation of the JOIN action so the model must infer that the marbles now are the superset and the qualifiers yesterday and today serve as indicators of subsets. It is also of interest that the use of the word altogether (rather than the word now) may help children infer that the now-set is the superset.*
sensitive to the number of inferences that are required in each of these versions, the predictor equations from the regression analyses in Chapter 4 can verify these results. Figure 13 shows the equations that will predict solution probability for first and third grade students.

By running the model and obtaining scores for the MEMINF and REACT variables, these values can be plugged into the equations to obtain the model's predicted solution probabilities.\(^7\) Tables 24 and 25 present children's solution probability with the traditional wording from Riley & Greeno (1988), children's solution probability with the time-altered wording from Staub and Reusser (1992) and the computer model's predicted solution probabilities on both versions for grades 1 and 3, respectively.

\[
p_1 = \text{MEMINF}(-0.47) + \text{REACT}(0.36) + 1.99
\]
\[
p_3 = \text{MEMINF}(-0.08) + \text{REACT}(-0.12) + 1.20
\]

Figure 13.
Equations predicting probability of solution for first and third grade

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Model's Prediction</th>
<th>Staub Data</th>
<th>Riley Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>1.00</td>
<td>-</td>
<td>1.00</td>
</tr>
<tr>
<td>Time-Altered</td>
<td>0.74</td>
<td>0.63</td>
<td>-</td>
</tr>
</tbody>
</table>

\(^7\) MEMINF is a measure of the total processing load on working memory (i.e., the sum of the average number of concepts in memory plus the average number of inferences required) and REACT is a dichotomous variable indicating whether a previously instantiated arithmetic action must be suppressed in favor of another. A more complete description of these two variables is given in Chapter 4.
Table 25.
Predicted and observed 3rd grade probability of solutions for Change 1

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Model’s Prediction</th>
<th>Staub Data</th>
<th>Riley Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>1.00</td>
<td>-</td>
<td>1.00</td>
</tr>
<tr>
<td>Time-Altered</td>
<td>0.97</td>
<td>0.69</td>
<td>-</td>
</tr>
</tbody>
</table>

As expected, the model closely predicts the solution probability for the traditional wordings. In the first grade time-altered version, the model does predict a decrease in solution probability (74% success rate) but not as much as a decrease as Staub and Reusser found (63%). In the third grade, the discrepancy is even greater. In the third grade time-altered version, the model predicts that third graders will not be affected by the time alteration (97%), which contradicts the results of Staub and Reusser (69%). Clearly, there is a degree of difficulty in the time-altered problems that the model is unable to account for. One possible reason may be that some children in the Staub study are confused with the difference between *today* and *now*, where the model is not. If this were the case, a number of children would be expected to give the answer of three (the number Alex had *today*). Staub reports, however, that only a small percentage of children who get it wrong give this answer (personal communication). Another possible reason from the cognitive modeling 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 the set [Alex has ? now] is the superset would appear to be a complex one; i.e., the word "now" is the only 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.

---

8 The model is expected to closely predict the solution probabilities on the traditional wordings because the predictor equations were derived from childrens’ 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.
by ownership (John has, Mary has, John and Mary have) or other features such as color (red marbles, blue marbles, marbles). Because the model counts all inferences with the same weight (i.e., +1), the model 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 the first indication that the types of inferences may be at least as important as, if not more important than the number of inferences.

In summary, there are a number of potential features which can be altered to make the traditional Change I problem easier or harder. Words which facilitate the integration of text, the use of related coactors and a consistent narrative focus can make this problem easier. Removing the time relation language and altering the time sequence can make this problem harder. Table 26 summarizes the four various wording types. The four problems are listed in an order of difficulty from easiest to most difficult as established by empirical studies and/or predicted by the computer model. Within each type, a multiple number of alternate rewordings are possible which add to the total number of possible versions. Words in parentheses, e.g., "(then)," are optional. Words in parentheses and bold, e.g., "(more, altogether)," are words or phrases which can be inserted to facilitate the relation between sets. Words in parentheses and italicized, e.g., "(In the beginning, now)," are words or phrases which can be inserted to increase the explicitness of time relations. Words in brackets, e.g., "[marbles]," are optional ellipsed phrases. For example, after the initial mention of marbles, the word may be removed from subsequent sentences. Words or phrases in brackets, e.g., "[his sister/Bethany]," refer to the option of [related/unrelated] coactors. It is hypothesized that within each of the four versions solution probability can increase by including (i) time relation language, (ii) set relation language and/or (iii) related coactors. These three factors, as well as (iv) a consistent narrative focus and (v) sequential ordering of events, are hypothesized to increase solution probability for all problems with significant action language. A detailed discussion of the relative ordering for other action problems in the benchmark set of 18 problems is given in Appendix C. In addition to the five rewording factors just discussed, Appendix C introduces the use of other specific
Table 26. Change 1 (result unknown-JOIN) worded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time; Consistent Protagonist</td>
<td>(In the beginning) Alex had 5 soda cans. (Then) Alex got 3 (more) [(soda) cans] from [(his sister/Bethany)]. How many [(soda) cans] does Alex have (now, altogether)?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; InConsistent Protagonist</td>
<td>(In the beginning) Alex had 5 soda cans. (Then) [(his sister/Bethany) gave 3 (more) [(soda) cans] to him. How many [(soda) cans] does Alex have (now, altogether)?</td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time; Consistent Protagonist</td>
<td>Today Alex got 3 soda cans from [(his sister/Bethany). Yesterday Alex had 5 soda cans. How many soda cans does Alex have (now, altogether)?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; InConsistent Protagonist</td>
<td>Today [(Alex’s sister/Bethany) gave 3 soda cans to Alex. Yesterday Alex had 5 soda cans. How many soda cans does Alex have (now, altogether)?</td>
</tr>
</tbody>
</table>

wordings in action problems to explicitly indicate (vi) the relationship between a source set and the amount given away (e.g. “gave 3 of them”), (vii) a direct modeling strategy (e.g., “how many do you need to get”) and (viii) the resulting set of an action (e.g., “has 2 (left, altogether”).

The next sections introduce the class of problems which neither contain significant actions nor relational language.

5.3.3. Making Static Non-Relational Problems Easier

Figure 14 shows Riley et al.’s (1988) Combine 5 problem. This problem is typical of a class of problems which do not contain significant actions (thus the term, static) and do not involve relational expressions such as “more than” (thus the term, non-relational). In Combine 5 problems, the first sentence describes a superset with a known amount. A subset with a known quantity is then described.
David and Kathy have 8 soda cans altogether. David has 5 soda cans. How many soda cans does Kathy have?

Figure 14.
Traditional Combine 5

followed by a question requesting the amount of the other subset. This problem has received considerable attention in the literature, mostly due to its level of difficulty. Considering all eighteen problems in Riley’s (1988) benchmark set, 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, 1988) studies reflect the difficulty: 22%-K, 33%-1st, 55%-2nd and 75%-3rd. These relatively low probabilities in even the higher grades are confirmed elsewhere (Davis-Dorsey et al., 1991; DeCorte, Verschaffel et al., 1985).

There are three potential rewordings which might help those students who are not able to solve this type of problem: (i) the use of related co-actors (e.g., David and his mother), (ii) the use of language to facilitate text integration between sentences and thereby highlighting the relationship between quantities (e.g., of them, the rest) and (iii) removal of the word altogether in combination with the suggestion in (ii).

The first alternative to the traditional wording, i.e., of using related co-actors, is discussed above.

The second potential alternative of Combine 5 is to include words which facilitate the process of text integration by highlighting the relationship between quantities, as shown in Figure 15. As discussed in Chapter 3 in the context of the computer model, the phrase of them in the second sentence generates an explicit reference to a previously mentioned set, i.e., David and Kathy’s set of 8. This explicit reference identifies that David’s 5 are part of the previously mentioned set of eight. In the traditional wording,

---

* Combine 6 varies from Combine 5 in that Combine 6 includes a sentence describing Kathy’s unknown set of some. While this added sentence does not add any new information, the computer model is sensitive to the change in the sense that it creates an increased memory load. This change from Combine 5 to 6 is discussed more fully in the context of making Combine 5 harder.
David and Kathy have 8 soda cans altogether. David has 5 of them. The rest [of them] are Kathy’s. How many soda cans does Kathy have?

Figure 15.
Including of them and the rest to make Combine 5 easier

"David has 5 soda cans," an inference is required to establish that David’s five cans are from David and Kathy’s original set of eight cans, as opposed to five totally unrelated cans.

In addition, the phrase "The rest" in the reworded version marks Kathy’s set of "some" as "those soda cans that are left as the result of a previous separation." In the traditional wording, no reference is made to Kathy’s set of "some" so the computer model must infer that Kathy’s cans are those remaining after separating out the cans belonging to David.

DeCorte, Verschaffel et al. (1985) and others have provided empirical support that the reworded version significantly increases children’s solution probability. In line with these results, the computer model predicts that including the phrases "of them" and "the rest" will increase solution probability. Table 27 compares first and second grade children’s success on both the traditional and reworded versions as reported by DeCorte, Verschaffel et al. (1985) and predicted by the computer model. For both grades (1st and 2nd), the model predicts a greater increase in solution probability (from the traditional to the reworded version) than found by DeCorte et al. As discussed in Chapter 3, the memory loads for these two problems are quite similar although the model performs three fewer inferences on the reworded version. As expected, the model’s predictor equation for first grade is more sensitive to a change in the number of inferences than a later grade.

The third potential rewording involves the use of the of them and the rest wordings as well as the removal of the word altogether in the first sentence, as shown in Figure 16. Cummins (1991) used this
Table 27.
Predicted and observed 1st and 2nd grade probabilities for reworded versions of Combine 5

<table>
<thead>
<tr>
<th>Problem Wording (grade)</th>
<th>Model’s Prediction</th>
<th>DeCorte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional (1st)</td>
<td>0.47</td>
<td>0.43</td>
</tr>
<tr>
<td>altogether &amp; of them (1st)</td>
<td>0.94</td>
<td>0.57</td>
</tr>
<tr>
<td>Traditional (2nd)</td>
<td>0.52</td>
<td>0.71</td>
</tr>
<tr>
<td>altogether &amp; of them (2nd)</td>
<td>0.81</td>
<td>0.83</td>
</tr>
</tbody>
</table>

There are 8 soda cans. 5 of them belong to David. The rest are Kathy’s. How many soda cans does Kathy have?

Figure 16.
Including set reference language and removing altogether in Combine 5

version in her studies of the factors that influence childrens’ interpretations of arithmetic word problems. The critical difference between this problem and the preceding version is the absence of the word altogether in the first sentence. According to Cummins (1991), the word altogether 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 validates her hypothesis with results from a computer simulation model, childrens’ recall protocalls (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. DeCorte, Verschaffel et al. (1985) published similar results. In short, removing the word altogether may avoid the potential misinterpretation that altogether means each.

The computer model also predicts that the Cummins (1991) version will be easier than the traditional wording, although the model offers a different rationale. As described in Chapter 3, the parser performs three tasks upon reading the word altogether: (i) partition the current set, although the qualifier which causes the partition (e.g., different owners, different colors) is currently unknown; (ii) attempt to determine the qualifying dependency that partitions the set; and (iii) instantiate the arithmetic action Join.10 In the traditional wording, after the first sentence is read, the computer model has the following representation in memory: {David & Kathy - 8 soda cans, JOIN}. While the “sets to join” are still unknown, the critical point is that the Join action has been activated. Upon reading the second sentence, the computer model 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 computer model reads the traditional wording, the Join action must be suppressed in favor of the Separating-From action. Because a significant percentage of children’s errors on the traditional version are arithmetic errors (i.e., they add 8 and 3 rather than subtracting 3 from 8), the model suggests that young children may not be able to suppress previously activated information, thus they remain with the initial Join activation. Given the Cummins (1991) reworded version without the presence of altogether in the initial sentence, the model is not required to suppress an initial activation of Join. Thus, the word “altogether” can both facilitate and hinder problem solving success, depending on its location in the problem text and a student’s (mis)understanding of the word. Table 28 compares first grade children’s success on both the

---

10 Recall that 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.
traditional and reworded versions as reported by Cummins (1991) and DeCorte, Verschaffel et al. (1985) and predicted by the computer model. Note especially that the model predicts less of an increase in solution probability (from the traditional to the of them only version) than reported by Cummins (1991), whereas the model predicts higher values than those reported by DeCorte. The model's prediction that 82% of first graders should be able to solve the of them only version is remarkable similar to the 85% reported by Cummins.

5.3.4. Making Static Non-Relational Problems Harder

There exist two potential rewordings which can make the traditional Combine 5 problem harder: (i) increase the level of abstraction of the semantic relation between the disjoint sets and (ii) add a sentence describing the unknown as a set of some. The first of these alternatives is discussed in Appendix C for Combine 1 and 2 problems.

The second alternative wording is the transition from Combine 5 to Combine 6. Figure 17 shows Riley et al.'s Combine 6 problem; the sentence in italics highlights the addition to Combine 5. The hypothesis that Combine 6 is more difficult than Combine 5 is based on the predictions of the computer model. As discussed in Chapter 3, the extra sentence in Combine 6 increases the average load on

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Model's Prediction</th>
<th>DeCorte Data</th>
<th>Cummins Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>0.47</td>
<td>0.43</td>
<td>0.30</td>
</tr>
<tr>
<td>altogether &amp; of them</td>
<td>0.94</td>
<td>0.57</td>
<td>-</td>
</tr>
<tr>
<td>of them only</td>
<td>0.82</td>
<td>-</td>
<td>0.85</td>
</tr>
</tbody>
</table>

Table 28. Predicted and observed 1st grade probabilities for reworded versions of Combine 5
David and Kathy have 8 soda cans altogether. Kathy has some soda cans. David has 5 soda cans. How many soda cans does Kathy have?

Figure 17.
Traditional Combine 6

working memory, as described next. In Combine 6, like Combine 5, the model determines that “Kathy’s some” are part of “David and Kathy’s 8” and thus should be SEPARATED-FROM the set of eight. The difference in memory load is evident in the model’s representation of the first two sentences of Combine 5 and 6:

Combine 5: [SEPARATE] [3 FROM 8]
Combine 6: [SEPARATE] [some] [FROM 8]

The rationale for the difference is that if two sets with known amounts (two concepts) are related with an arithmetic action (one concept) then those three concepts are merged into two concepts, as shown for Combine 5. In Combine 6, however, Kathy’s set of some does not have a known quantity so “some FROM 8” is not merged into one concept. In Combine 6, the model must “remember” three concepts as it begins to process the third sentence. In terms of predicted scores, Table 29 compares the model’s predicted scores for Combine 5 and 6 and the associated probabilities as determined by Riley et al. (1988). Riley et al. (1988) found that first, second and third graders did at least as well or better on Combine 6 than on Combine 5, counter to the predictions of the model, as shown in Table 29. One potential explanation is that the extra memory load caused by the additional sentence describing the unknown set does not adversely affect a number of children, especially those who are beginning to appreciate the critical part-part-whole concept. Another, perhaps more simple, explanation is that the extra sentence focuses attention on the unknown set such that it serves as a convenient placeholder.
Table 29. 
Predicted and observed probabilities of solution for Combine 5 and 6

<table>
<thead>
<tr>
<th>Grade Probability</th>
<th>Predicted Combine 5</th>
<th>Riley Combine 5</th>
<th>Predicted Combine 6</th>
<th>Riley Combine 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_k )</td>
<td>0.27</td>
<td>0.22</td>
<td>0.09</td>
<td>0.17</td>
</tr>
<tr>
<td>( p_1 )</td>
<td>0.47</td>
<td>0.33</td>
<td>0.24</td>
<td>0.44</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>0.52</td>
<td>0.55</td>
<td>0.38</td>
<td>0.55</td>
</tr>
<tr>
<td>( p_3 )</td>
<td>0.75</td>
<td>0.75</td>
<td>0.71</td>
<td>0.85</td>
</tr>
</tbody>
</table>

The majority of children in the early grades experience some difficulty with both Combine 5 and 6 versions. Even at the second grade level, only about half of the children in Riley’s study solve these problems correctly. Table 30 summarizes the hypothesized ordering of problem wordings relative to the traditional wordings of Combine 5 and 6. The effects of these rewording factors on other non-relational problems are detailed in Appendix C.

5.3.5. Making Static Relational Problems Easier

Figure 18 shows Riley et al.’s (1988) Compare 3 problem. This problem is typical of a class of problems which do not contain significant actions (thus the term, static) and do involve relational expressions such as “more than.” In Compare 3 problems, a set with an unknown amount (referred to as the compared set) is related to a previously mentioned set with a known amount (referred to as the referent set). The compared set is described as being in excess of (i.e., more than) the referent set. The question requests the amount of the compared set.

Figure 19 shows Riley’s (1988) Compare 4 problem. In Compare 4 problems, like Compare 3, a compared set with an unknown amount is related to a referent set with a known amount. In Compare 4, however, the compared set is described as being less than the referent set. The question requests the
Table 30.
Combine 5 and 6 (subset unknown--SEPARATE-FROM) reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>No altogether; Explicit Part; Explicit Result</td>
<td>There are 8 soda cans. 5 of them belong to David. The rest of them are [Kathy's/his mother's]. How many [(soda) cans] does [Kathy/his mother] have?</td>
</tr>
<tr>
<td></td>
<td>Explicit Part Of; Explicit Result</td>
<td>David and [Kathy/his mother] have 8 soda cans (altogether). 5 of them belong to David. The rest of them are [Kathy's/his mother's]. How many [(soda) cans] does [Kathy/his mother] have?</td>
</tr>
<tr>
<td></td>
<td>Explicit Part Of</td>
<td>David and [Kathy/his mother] have 8 soda cans (altogether). 5 of them belong to David. How many [(soda) cans] does [Kathy/his mother] have?</td>
</tr>
<tr>
<td></td>
<td>Traditional 5 Wording</td>
<td>David and [Kathy/his mother] have 8 soda cans (altogether). David has 5 [(soda) cans]. How many [(soda) cans] does [Kathy/his mother] have?</td>
</tr>
<tr>
<td></td>
<td>Traditional 6 Wording</td>
<td>David and [Kathy/his mother] have 8 soda cans (altogether). [Kathy/his mother] has some [(soda) cans]. David has 5 [(soda) cans]. How many [(soda) cans] does [Kathy/his mother] have?</td>
</tr>
<tr>
<td>Hardest</td>
<td>Action Differentiation</td>
<td>8 children went to school (altogether)? Some [children] ran to school. 3 [children] walked to school. How many [children] ran to school?</td>
</tr>
</tbody>
</table>

Jacob has 3 soda cans. Chrissy has 2 soda cans more than Jacob. How many soda cans does Chrissy have?

Figure 18.
Traditional Compare 3

amount of the compared set.
Jacob has 3 soda cans. Chrissy has 2 soda cans less than Jacob. How many soda cans does Chrissy have?

Figure 19.
Traditional Compare 4

The two alternate wordings (more than with known referent set, less than with known referent set) are referred to as consistent Compare problems according to the language of Lewis and Mayer (1987). These two problems are consistent because the language used to describe the relation is consistent with the necessary arithmetic operation in each problem. For example, Compare 3 with more is "consistent" since that problem requires addition and more "implies" addition. Compare 4 with less is "consistent" since that problem requires subtraction and less "implies" subtraction. Consistent language problems are in contrast to problems with inconsistent language; i.e., the problem says more but requires subtraction or says less but requires addition. Inconsistent language problems have been convincingly shown to be more difficult than consistent language problems (Hegarty et al., 1992, Lewis & Mayer, 1987; Stern, 1993, Verschaffel et al., 1992). These results confirm that the consistent Compare 3 and 4 problems described in this section are easier than inconsistent Compare 5 and 6 problems presented in the next section.

The relative difference in difficulty between the two consistent problems, Compare 3 and 4, is not as straightforward. The observed solution probabilities for these two problems across a number of studies show that children perform about the same on both problems, as summarized in Table 31. The results from the eye movement experiments of Verschaffel et al. (1992) show the same amount of arithmetic reversal errors (e.g., children use addition when subtraction is needed) for Compare 3 as Compare 4 (3.3% in each case). On the other hand, in the eye movement experiments of Hegarty et al. (1992), college students with high accuracy scores took a greater amount of time to solve consistent problems containing lexically "marked" terms (e.g., less) than to solve consistent problems containing "unmarked"
Table 31.
Comparison of Average Solution Results Across Grades for Compare 3 and 4 Problems

<table>
<thead>
<tr>
<th>Study (grades)</th>
<th>Compare 3</th>
<th>Compare 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Okamoto (1992) (K-4)</td>
<td>0.47</td>
<td>0.50</td>
</tr>
<tr>
<td>Rathsell (1986) (2-5)</td>
<td>0.79</td>
<td>0.80</td>
</tr>
<tr>
<td>Riley (1983) (K-3)</td>
<td>0.48</td>
<td>0.43</td>
</tr>
<tr>
<td>Riley (1988) (K-3)</td>
<td>0.48</td>
<td>0.50</td>
</tr>
</tbody>
</table>

terms (e.g., more). However, the total time results did not extend to the low accuracy group, that is, low accuracy students spent equal amounts of time on Compare 3 and 4 problems.

The discrepancy between low and high accuracy students in the Hegarty et al. (1992) data suggests that for successful students, the problems with marked terms (e.g., less) involve additional processing that is not required for problems with unmarked terms (e.g., more). On the other hand, Walkerdine (1990) suggests that children's difficulties with words such as less are due to a lack of social use of those words. That is, Walkerdine claims that words such as less neither have more complex lexical entries nor do they require additional processing as compared to other relational words such as more. Having examined the corpus of words used in home interactions between mothers and their young children, Walkerdine found that "more" appears many times and "less" does not appear even once:

... more is the positive term, though its value does not come from the internal relations of the linguistic system or a set of perceptual universals; it comes from the regulation of the social practices which make up our culture (p. 27).

In short, the traditional wordings of Compare 3 and 4 are essentially equal in difficulty, although young children may be more familiar with the more than wording in Compare 3. In addition, the
computer model predicts that these two problems involve identical loads on working memory and an equal number of inferences.

There exist two possibilities to make the traditional wording of Compare 3 and 4 easier: the use of (i) Equalize language and (ii) related co-actors. With Equalize language, the static relational language (e.g., how many more/less than) in the traditional wordings of Compare 3 and 4 can be replaced with action language which describes a hypothetical act of making the two sets equal (e.g., how many does one need to find/lose to have as many as the other), as described above for Compare 1 and 2. For all grade levels, the computer model predicts that the Equalize version will improve solution probability, especially for the youngest grades. As explained previously, the Equalize version provides an explicit action to Join and does not require the inferences that are necessary when the model solves the situationally void traditional wording. Table 32 compares the solution probabilities of the traditional Compare 3 as found by Riley et al. (1988), the model's predicted solution probabilities of the traditional wording and the model's predicted probabilities on the Equalize version. The results for the Equalize version of Compare 4 are similar.

<table>
<thead>
<tr>
<th>Solution Probability</th>
<th>Traditional Wording (Riley data)</th>
<th>Traditional Wording (predicted)</th>
<th>Equalize Wording (predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p_1 )</td>
<td>0.09</td>
<td>0.20</td>
<td>0.56</td>
</tr>
<tr>
<td>( p_2 )</td>
<td>0.33</td>
<td>0.42</td>
<td>0.89</td>
</tr>
<tr>
<td>( p_3 )</td>
<td>0.60</td>
<td>0.71</td>
<td>1.00</td>
</tr>
<tr>
<td>( p_4 )</td>
<td>0.90</td>
<td>0.92</td>
<td>1.00</td>
</tr>
</tbody>
</table>
5.3.6. Making Static Relational Problems Harder

There currently is no hypothesis for making these problems more difficult. At present, this seems reasonable since these problems represent a very high level of mathematical abstraction, i.e., the text is void of any situational language or language that implies the results of situations (e.g., left, altogether). Table 33 summarizes a relative order of difficulty for the more-than and less than (compared quantity unknown) types of problems.

The effects of the equalizing rewording on other relational problems are detailed in Appendix C. In addition, Appendix C details the specific rewordings which replace the static relational language (e.g., "how many more than") in difference-unknown problems with action language which facilitates a one-

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>&quot;If Find&quot;;</td>
<td>Jacob has 3 soda cans. If he finds 5 (more) [(soda) cans], he will have the same number [of (soda) cans] as [Chrissy/his cousin]. How many [(soda) cans] does [Chrissy/his cousin] have?</td>
</tr>
<tr>
<td></td>
<td>&quot;As Many As&quot;</td>
<td></td>
</tr>
<tr>
<td>&quot;If Lose&quot;;</td>
<td>&quot;As Many As&quot;</td>
<td>Chrissy has 8 soda cans. If she loses 5 (of them, [(soda) cans]), she will have the same number [of (soda) cans] as [Jacob/her cousin]. How many [(soda) cans] does [Jacob/her cousin] have?</td>
</tr>
<tr>
<td>&quot;More Than&quot;;</td>
<td>Traditional</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 more [(soda) cans] than Jacob. How many [(soda) cans] does [Chrissy/his cousin] have?</td>
</tr>
<tr>
<td>Traditional</td>
<td>Compare 3</td>
<td></td>
</tr>
<tr>
<td>hardest</td>
<td>&quot;Less Than&quot;;</td>
<td>Chrissy has 8 soda cans. [Jacob/Her cousin] has 5 less [(soda) cans] than Chrissy. How many [(soda) cans] does [Jacob/her cousin] have?</td>
</tr>
<tr>
<td></td>
<td>Traditional</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Compare 4</td>
<td></td>
</tr>
</tbody>
</table>
to-one correspondence matching strategy (e.g., "how many won't get").

5.4. Summary

The preceding sections offer a detailed first pass at extending the traditional classification of word problems. Each individual section presented a multiple number of ways to alter traditional wordings, including the use of (i) words which facilitate text integration, (ii) action language, (iii) a consistent narrative focus, (iv) related co-actors and (v) "mathematical" words which highlight the role of a particular set such as superset or the result of a particular arithmetic action. A relative ordering of difficulty for the multiple wordings of each type of problem was determined by previous results where available and/or by predictions from the computer model.

The larger purpose of this chapter is to propose an integrated network of problems which offers both classroom teachers and a future learning environment new alternatives for assessment and/or for choosing the "next best problem" to present. As Carpenter et al. (1988) have pointed out, teachers' instruction can improve when they begin with a coherent knowledge of problem types and the problem solving strategies children use to solve those problems. Beyond the relative ordering of difficulty for each problem type, the network can suggest a number of problems which address a topic in the elementary mathematics curriculum. For example, imagine a database where all word problems are "potentially" linked or "networked" together. A search request could be made (e.g., from the teacher to the database) for a list of all problems involving "addition when the superset is unknown." This parameter crosses the traditional boundaries of semantic structure. The network could suggest a variety of semantic types of problems as well as multiple ways to reword each type. The traditional analyses based on semantic structure offer only the following ordering of difficulty: Change I < Combine I < Compare I. On the other hand, when the parameter, addition with superset unknown, is mapped onto the network of problems, a significantly larger subset of problems is established, each ordered according to predicted levels of difficulty. Table 34 shows such a subset.
Table 34.
Subset of problems from the network for “addition with the superset unknown”

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Explicit Transfer-In;</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em></td>
</tr>
<tr>
<td></td>
<td>Sequential time;</td>
<td>Alex got 3 <em>(more)</em> <em>(soda) cans</em> from <em>(his sister/Bethany)</em>. How many <em>(soda) cans</em> does Alex have <em>(now, altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Consistent Protagonist</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Explicit Transfer-In;</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em></td>
</tr>
<tr>
<td></td>
<td>Sequential time;</td>
<td>*(his sister/Bethany) gave 3 <em>(more)</em> <em>(soda) cans</em> to Alex. How many <em>(soda) cans</em> does Alex have <em>(now, altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Inconsistent Protagonist</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hypothetical Join action</td>
<td>Alex has 5 soda cans. *(His sister/Bethany) has 3 <em>(soda) cans</em>. If <em>(Alex and (his sister/Bethany), they)</em> put their <em>(soda) cans</em> together, how many <em>(soda) cans</em> will <em>(Alex and (his sister/Bethany), they)</em> have <em>(altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Equalizing Join action</td>
<td>Alex has 3 soda cans. If he finds 5 <em>(more)</em> <em>(soda) cans</em>, he will have the same number <em>(of soda cans)</em> as <em>(his sister/Bethany)</em>. How many <em>(soda) cans</em> does <em>(his sister/Bethany)</em> have?</td>
</tr>
<tr>
<td></td>
<td>Implied Join action</td>
<td>David has 5 soda cans. *(His mother/Kathy) has 3 <em>(soda) cans</em>. How many <em>(soda) cans</em> do <em>(David and (his mother/Kathy), they)</em> have <em>(altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Explicit Transfer-In;</td>
<td>Today <em>(Alex’s sister/Bethany)</em> gave 3 soda cans to Alex. <em>(Alex’s sister/Bethany)</em> yesterday had 5 soda cans. How many soda cans does Alex have <em>(now, altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Non-Sequential time;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Consistent Protagonist</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Explicit Transfer-In;</td>
<td>Today <em>(Alex’s sister/Bethany)</em> gave 3 soda cans to Alex. <em>(Alex’s sister/Bethany)</em> yesterday had 5 soda cans. How many soda cans does Alex have <em>(now, altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Non-Sequential time;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inconsistent Protagonist</td>
<td></td>
</tr>
<tr>
<td>hardest</td>
<td>Static Comparison</td>
<td>Alex has 3 soda cans. *(His sister/Bethany) has 5 <em>(soda) cans</em> more than *(Alex). How many <em>(soda) cans</em> does *(his sister/Bethany) have?</td>
</tr>
</tbody>
</table>

In short, the computer model’s focus on the processes of reading leads to new insights into the nature of problem difficulty. A network of problems such as presented in this chapter offers classroom
teachers with multiple ways to rephrase problems to help the frustrated reader and challenge the more advanced readers.
CHAPTER 6

Conclusion

This research provides new information processing evidence concerning the sources of difficulty that children experience when solving arithmetic word problems. A computer model simulates the cognitive requirements facing a young reader during sentence-level reading and text integration. The system design of the computer model and associated results on problem difficulty have concrete implications for:

1. the development of a total model of the problem solving process, including the various difficulties involved in converting from natural language statements to representations of a problem solution;

2. curriculum development and assessment; and

3. the use of computer-based learning environments in the classroom.

6.1. The Importance of Starting at the Beginning

The computer model of arithmetic word problem solving is unique in that it begins the problem solving process with the same form of problem representation as presented to children. This design decision forced us to consider not only the critical role of individual words (e.g., "more" can be interpreted as "in addition to"), but also a specification of the possible stages of representation that are traversed between natural language and a symbolic representation such as an equation. In particular, the simulation models a novice problem solver who reads in a line-by-line, "bottom-up" fashion and works toward a procedural interpretation of the problem. Procedural interpretations are derived from either the actions in the text or by inference through the process of text integration. In order to highlight the tacit processes that are simulated during comprehension, the model monitors specific cognitive demands as it makes the transition between increasingly abstract representations. Children’s probability of solution was analyzed in regression analyses as a function of text comprehension processes. A measure of the total processing
demand on working memory (a combination of the average number of conceptual units to remember and the average number of inferences) accounted for a significant amount of variance in children’s probability of solution across four grade levels (K-3). This result provides a new focus on the differences in information processing between problems which vary by traditional features such as semantic structure and location of the unknown (e.g., why Referent-Set-Unknown (Compare 5&6) problems are more difficult than Compare-Set-Unknown (Compare 3&4) problems), as well as by differences based on slight changes in problem wording.

6.2. Implications for Education

As a result of the regression analyses presented in Chapter 4, the model helps create a systematic network of problems which vary according to parameters such as linguistic complexity and memory requirements. The network of problems is hypothesized to provide better-defined areas of focus in curriculum development and assessment.

6.2.1. Curriculum Development

A network of word problems as proposed in this work is one contribution to the ongoing changes in the content and emphasis of the elementary mathematics curriculum (grades K-4). Specifically, the NCTM Curriculum and Evaluation Standards (NCTM, 1989) call for increased attention in problem solving which includes:

(1) word problems with a variety of structures, and

(2) use of everyday problems.

The network addresses both of these points. First, the “early years” mathematics curriculum must advance beyond a point where word problems are an adjunct to computational exercises. If encouraging children to solve word problems is a positive “problem solving” exercise (and I believe it is), then children must be exposed to: (i) multiple versions of the same type of problem, and perhaps more
importantly, (ii) *multiple types* of problems. It is clearly not enough to continually expose children to isolated types of problems. For example, children who are repeatedly asked to solve word problems of one semantic type (e.g., Combine 1) are in fact implicitly encouraged to find problem solving "shortcuts."

With continued exposure to Combine 1 problems, it may not take long for children to infer that "all these problems use altogether and they ask me to add." Mathematics educators refer to such a shortcut as a "keyword strategy." In fact, the only item to receive *decreased attention* in the area of problem solving in the Standards (NCTM, 1989) is the use of "cluewords" to determine which mathematical operation to apply.

As shown in Chapter 5 and Appendix C, a network of problems begins to provide curriculum designers (and teacher educators and classroom teachers!) with the type of information they need to expose children to problems of multiple structures and wordings. Because there are many ways to describe mathematical relationships, the elementary mathematics education curriculum must expose children to multiple ways to describe those relationships.

6.2.2. Assessment

The refinement of problem types beyond the classification of semantic structure can increase teacher awareness of the *multiple* sources of problem difficulty. For each traditional problem wording, the network of problems suggests wordings to address both the needs of the students who are experiencing difficulties and those who need to be challenged. This has direct consequences for the assessment of children's problem solving skills. Using the proposed network of word problems, multiple wordings of the "same" problem provide an ability to differentiate problem solving tasks, e.g., reading comprehension vs. logico-mathematical knowledge. For example, consider a student who is unable to solve a particular problem. If a teacher finds that a particular rewording helps the student, that student's trouble may have more to do with the comprehension of an abstract mathematical expression rather than an inability to "perform the math." Prior to further assessment of a particular mathematical competence,
the teacher can provide practice with situations which can be described with the troublesome language. Likewise, a teacher can challenge gifted students by presenting reworded versions which require an increasing number of inferences on the student's part.

In short, the network provides a framework to assist teachers with their selection of the "next best" problem to present. Of course, a network as proposed here can become very large. In fact, I currently envision it as a database that a teacher would access rather than memorize. Future plans for a system which incorporates such an "electronic" aid for teaching and assessment is discussed in the next section.

6.3. Towards an Intelligent Learning Environment

The computer model offers exciting potential for the productive future use of an interactive learning environment in the classroom to help children become competent with mathematical language. In particular, the reading component suggests the following directions for future research. First, a learning environment which incorporates the reading component can offer intentionally limited guidance in the form of problem sequencing. As discussed above, the system can make hypotheses concerning the "next best problem" based on a student's problem solving history and potential rewordings. Second, the natural language capabilities of the model make possible a unique microworld tool in an interactive learning environment. Specifically, such a tool would allow the creation of a dynamic link between linguistic representations and other graphically displayed representations such as icon groupings and equations. In a prototype version of this interactive tool (LeBlanc, 1992), the conceptual representations produced by EDUCE and SELAH serve as the dynamic link between the verbal and iconic representations. Further development of this tool in the future could allow students to manipulate one kind of representation and see simultaneous effects on other representations. That is, if a student altered the wording in a sentence to reflect a new relationship, then this would lead to a new conceptual representation which in turn would lead to a new iconic or numeric representation. Likewise, a change in the iconic or numeric representation would cause a new sentence to appear which expresses the new relationship in natural language. For
example, the system could provide dynamic practice with complex relational language, e.g., "have more than." Students could thus be encouraged to appreciate the "mathematical meaning" of complex linguistic constructions; an awareness of the links between natural language and other representations could help the student to understand how one mathematical concept or representation can account for many linguistic expressions of everyday situations. A teacher would find it difficult if not impossible to provide such dynamic features effectively by moving back and forth between various representational media when the need arises for a particular child. The computer tools would complement the teacher's role by performing desirable but otherwise infeasible tasks.

And finally, the learning environment offers the opportunity for teachers to integrate writing and mathematics through computer solutions of student-created word problems. Given the current interest in the integration of writing with other disciplines, the ability to parse natural language could have an impact on the use of writing mathematics as a pedagogical tool. A typical use is envisioned as follows. Initially, students would write short stories which require mathematical problem solving. Teachers could then review these stories, commenting on structure, spelling, etc., and students would edit their work. The "reading" facility would then allow students to enter their arithmetic stories and "see" the learning environment (and other students!) attempt to solve their problem; a role reversal from the traditional situation. In doing so, students could use real-world associations and personalizations, e.g., the use of classmates' names, which may more fully engage the student.

6.4. Final Thoughts

The system design of a computer model to read, comprehend and solve arithmetic word problems serves as a classic example of cognitive science research. The implementation represents an interdisciplinary effort, combining techniques from artificial intelligence, cognitive psychology and mathematics education. The advantage of this approach is, of course, the integration of perspectives: mathematics education focuses on helping children learn, cognitive psychology encourages systematic experimentation and
artificial intelligence applies computational strategies to problems which do not easily lend themselves to algorithmic solutions. Specific to this work, EDUCE and SELAH expose tacit problem solving tasks while providing new explanations of problem difficulty which suggest a number of possibilities for instructional sequencing. The model provides a detailed look at the sources of a young word problem solver’s frustration or a teacher’s challenge, depending on one’s point of view.
APPENDIX A

Benchmark Set of 18 Word Problems

Change

(1) Jacob had 5 soda cans. Then Chrissy gave Jacob 3 soda cans. How many soda cans does Jacob have now? (Change 1)

(2) Jacob had 5 soda cans. Then Jacob gave Chrissy 3 soda cans. How many soda cans does Jacob have now? (Change 2)

(3) Jacob had 5 soda cans. Then Chrissy gave Jacob some soda cans. Now Jacob has 8 soda cans. How many soda cans did Chrissy give Jacob? (Change 3)

(4) Jacob had 5 soda cans. Then Jacob gave Chrissy some soda cans. Now Jacob has 2 soda cans. How many soda cans did Jacob give Chrissy? (Change 4)

(5) Jacob had some soda cans. Then Chrissy gave Jacob 3 soda cans. Now Jacob has 8 soda cans. How many soda cans did Jacob have in the beginning? (Change 5)

(6) Jacob had some soda cans. Then Jacob gave Chrissy 5 soda cans. Now Jacob has 3 soda cans. How many soda cans did Jacob have in the beginning? (Change 6)

Combine

(7) Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans do Jacob and Chrissy have altogether? (Combine 1)

(8) Jacob and Chrissy have some soda cans. Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans do Jacob and Chrissy have altogether? (Combine 2)

(9) Jacob has 5 soda cans. Chrissy has some soda cans. Jacob and Chrissy have 8 soda cans altogether. How many soda cans does Chrissy have? (Combine 3)

(10) Jacob has some soda cans. Chrissy has 5 soda cans. Jacob and Chrissy have 8 soda cans altogether. How many soda cans does Jacob have? (Combine 4)

(11) Jacob and Chrissy have 8 soda cans altogether. Jacob has 5 soda cans. How many soda cans does Chrissy have? (Combine 5)

(12) Jacob and Chrissy have 8 soda cans altogether. Chrissy has some soda cans. Jacob has 3 soda cans. How many soda cans does Chrissy have? (Combine 6)

Compare

(13) Jacob has 5 soda cans. Chrissy has 2 soda cans. How many soda cans does Jacob have more than Chrissy? (Compare 1)

(14) Jacob has 5 soda cans. Chrissy has 2 soda cans. How many soda cans does Chrissy have less than Jacob? (Compare 2)

(15) Jacob has 3 soda cans. Chrissy has 2 soda cans more than Jacob. How many soda cans does Chrissy have? (Compare 3)

(16) Jacob has 5 soda cans. Chrissy has 2 soda cans less than Jacob. How many soda cans does Chrissy have? (Compare 4)

(17) Jacob has 5 soda cans. He has 2 soda cans more than Chrissy. How many soda cans does Chrissy have? (Compare 5)

(18) Jacob has 5 soda cans. He has 2 soda cans less than Chrissy. How many soda cans does Chrissy have? (Compare 6)
APPENDIX B

Summary of SELAH's Text Integration Scores

This appendix summarizes the text integration scores used in Experiments 1 and 3. Specifically, for each of the 18 problems in the benchmark set, a summary is given which explains how the model 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 inferences that are made.

As SELAH solves each problem, it monitors the amount of units or “concepts” in working memory at the end of the sentence. 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]). 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.

8.1. Change 1

| Alex had 5 marbles. Then Bethany gave Alex 5 marbles. How many marbles does Alex have now? |
|---------------------------------------------------------------|------------------|------------------|------------------|------------------|
|                                                                 | 1                | 2                | 3                | Average          |
| Processing Summary                                           | make set         | transfer-in situation; explicit JOIN action | explicit result of JOIN |                  |
| Concepts in Memory                                            | [Alex’s 5]       | [JOIN Beth’s 3]  | [JOIN Beth’s 3]  |                  |
|                                                               | [TO Alex’s 5]    | [TO Alex’s 5]    | [RE Alex’s 2]    |                  |
| # of concepts                                                 | 1                | 2                | 3                | 2.00             |
| # of inferences                                               | 0                | 0                | 0                | 0.00             |
8.2. Change 2

| Alex had 5 marbles. Then Alex gave Bethany 3 marbles. How many marbles does Alex have now? |
|---|---|---|---|---|
| **Sentence** | 1 | 2 | 3 | Average |
| **Processing Summary** | make set | transfer-out situation; infer 3 part-of 5; explicit SEP-FROM action | explicit result of SEP-FROM |
| **Concepts in Memory** | [Alex’s 5] | [SEP Bethany’s 3] | [SEP Bethany’s 3] |
| | [FR Alex’s 5] | [FR Alex’s 5] | [RE Alex’s 3] |
| # of concepts | 1 | 2 | 3 | 2.00 |
| # of inferences | 0 | 1 | 0 | 0.33 |

8.3. Change 3

| Alex had 5 marbles. Then Bethany gave Alex some marbles. Now Alex has 8 marbles. How many marbles did Bethany give Alex? |
|---|---|---|---|---|---|
| **Sentence** | 1 | 2 | 3 | 4 | Average |
| **Processing Summary** | make set | transfer-in situation; explicit JOIN action | explicit result of JOIN; Bethany gave ? |
| **Concepts in Memory** | [Alex’s 5] | [JOIN] | [JOIN] |
| | [Alex’s same] | [Alex’s same] | [RE Alex’s 8] |
| | [TO Alex’s 5] | [TO Alex’s 5] | [RE Alex’s 8] |
| # of concepts | 1 | 3 | 4 | 4 | 3.00 |
| # of inferences | 0 | 0 | 0 | 0 | 0.00 |
### 8.4. Change 4

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make set</td>
<td>transfer-out situation; infer some part-of 5; explicit SEP-FROM action</td>
<td></td>
<td>explicit result of SEP-FROM;</td>
<td>Alex gave ?</td>
<td></td>
</tr>
<tr>
<td>Concepts in Memory</td>
<td>[Alex’s 5]</td>
<td>[SEPARATE]</td>
<td>[SEPARATE]</td>
<td>[SEPARATE]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Beth’s some]</td>
<td>[Beth’s some]</td>
<td></td>
<td>[FR Alex’s 5]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[FR Alex’s 5]</td>
<td></td>
<td></td>
<td>[RE Alex’s 2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RE Alex’s 5]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
<td></td>
</tr>
</tbody>
</table>

### 8.5. Change 5

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make set</td>
<td>transfer-in situation; explicit JOIN action</td>
<td>explicit result of JOIN</td>
<td>Alex had ?</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Concepts in Memory</td>
<td>[Alex’s some]</td>
<td>[JOIN]</td>
<td>[JOIN]</td>
<td>[JOIN]</td>
<td>[JOIN]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Alex’s 3]</td>
<td>[Alex’s 3]</td>
<td>[TO Alex’s some]</td>
<td>[TO Alex’s some]</td>
<td>[TO Alex’s ?]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[TO Alex’s some]</td>
<td></td>
<td></td>
<td>[RE Alex’s 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RE Alex’s 8]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00</td>
<td></td>
</tr>
</tbody>
</table>
### 8.6. Change 6

**Alex had some marbles. Then Alex gave Bethany 3 marbles.**

**Now Alex has 2 marbles. How many marbles did Alex have in the beginning?**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set</td>
<td>transfer-out situation; infer 3 part-of <em>some</em>; explicit SEP-FROM action</td>
<td>explicit result of SEP-FROM;</td>
<td>Alex had?</td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[Alex’s <em>some</em>] [Bethany’s 3] [FR Alex’s <em>some</em>]</td>
<td>[SEPARATE] [Bethany’s 3] [FR Alex’s <em>some</em>] [RE Alex’s 2]</td>
<td>[SEPARATE] [Bethany’s 3] [FR Alex’s ?] [RE Alex’s 2]</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>3.00</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.25</td>
</tr>
</tbody>
</table>

### 8.7. Combine 1

**Alex has 5 marbles. Bethany has 3 marbles.**

**How many marbles do they have altogether?**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set</td>
<td>make set</td>
<td>infer JOIN; They have?</td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[Alex’s 5] [Beth’s 3]</td>
<td>[Alex’s 5] [Beth’s 3]</td>
<td>[JOIN Beth’s 3] [TO Alex’s 5] [RE their ?]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2.00</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.33</td>
</tr>
</tbody>
</table>
### 8.8. Combine 2

**Alex and Bethany have some marbles. Alex has 5 marbles. Bethany has 3 marbles. How many marbles do they have altogether?**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set</td>
<td>make set</td>
<td>make set</td>
<td>infer JOIN: they have?</td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[A&amp;B’s <em>some</em>]</td>
<td>[A&amp;B’s <em>some</em>]</td>
<td>[A&amp;B’s <em>some</em>]</td>
<td>[JOIN Beth’s 3]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Alex’s 5]</td>
<td>[Alex’s 5]</td>
<td>[Beth’s 3]</td>
<td>[TO Alex’s 5]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>2.25</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0.25</td>
</tr>
</tbody>
</table>

### 8.9. Combine 3

**Alex has 5 marbles. Bethany has some marbles. Alex and Bethany have 8 marbles altogether. How many marbles does Bethany have?**

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set</td>
<td>make set</td>
<td>make set; infer JOIN; infer 5 part-of 8; infer <em>some</em> part-of 8</td>
<td>Bethany has 2</td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[Alex’s 5]</td>
<td>[Alex’s 5]</td>
<td>[JOIN]</td>
<td>[JOIN]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Bethany’s <em>some</em>]</td>
<td>[Bethany’s <em>some</em>]</td>
<td>[TO Alex’s 5]</td>
<td>[TO Alex’s 5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RE A&amp;B’s 8]</td>
<td>[RE A&amp;B’s 8]</td>
<td>[RE A&amp;B’s 8]</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2.75</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.75</td>
</tr>
</tbody>
</table>
8.10. Combine 4

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set</td>
<td>make set</td>
<td>make set:</td>
<td>Alex has ?</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>infer JOIN;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>infer <em>some</em> part-of 8;</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>infer 3 part-of 8</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[Alex's <em>some</em>]</td>
<td>[Alex's <em>some</em>]</td>
<td>[JOIN]</td>
<td>[JOIN]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[Bethany's 3]</td>
<td>[TO Alex's <em>some</em>]</td>
<td>[Bethany's 3]</td>
<td>[TO Alex's?]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>[RE A&amp;B's 8]</td>
<td>[RE A&amp;B's 8]</td>
<td>[RE A&amp;B's 8]</td>
<td>[RE A&amp;B's 8]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>2.75</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0.75</td>
</tr>
</tbody>
</table>

8.11. Combine 5

<table>
<thead>
<tr>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set;</td>
<td>make set;</td>
<td>make set (B's ?);</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer JOIN</td>
<td>infer 5 part-of 8;</td>
<td>infer ? part-of 8;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>infer SEP-FROM;</td>
<td>infer SEP-FROM;</td>
<td>infer ? result of SEP-FROM</td>
<td></td>
</tr>
<tr>
<td></td>
<td>REACT</td>
<td>REACT</td>
<td>REACT</td>
<td></td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[A&amp;B's 8]</td>
<td>[SEP Alex's 5]</td>
<td>[SEP Alex's 5]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>JOIN</td>
<td>[FR A&amp;B's 8]</td>
<td>[FR A&amp;B's 8]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[RE Beth's?]</td>
<td>[RE Beth's?]</td>
<td></td>
</tr>
<tr>
<td># of concepts</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.33</td>
</tr>
<tr>
<td># of inferences</td>
<td>1</td>
<td>2</td>
<td>2</td>
<td>1.67</td>
</tr>
</tbody>
</table>
8.12. Combine 6

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>make set; infer JOIN</td>
<td>make set; infer <em>some</em> part-of 8; infer SEP-FROM; REACT</td>
<td>Beth has?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>Sentence</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>[A&amp;B’s 8] JOIN</td>
<td>SEPARATE [Beth’s same] [FR A&amp;B’s 8]</td>
<td>SEPARATE [Beth’s ?] [FR A&amp;B’s 8] [RE Alex’s 5]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of concepts</th>
<th># of inferences</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3.25</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>4</td>
<td>2</td>
<td>1.25</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

8.13. Compare 1

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>make set</td>
<td>make set (Alex has how many); referent set found; infer JOIN; infer Alex also has an amount equal to B’s 3; infer set of 8 is result of action</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>Sentence</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Alex’s 8] [Beth’s 3]</td>
<td>JOIN [Alex’s ?] [TO (Alex’s 3)] [RE Alex’s 8]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of concepts</th>
<th># of inferences</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>1.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>2.33</td>
<td></td>
</tr>
</tbody>
</table>
8.14. Compare 2

Alex has 8 marbles. Bethany has 3 marbles.
How many marbles does Bethany have less than Alex?

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make set (B has qty which is how many less than A has); referent set found; infer SEP-FROM; infer transformation of &quot;how many less 8&quot; to &quot;8 less how many&quot;; infer Beth's 3 are result of action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>[Alex's 8]</th>
<th>[Alex's 8]</th>
<th>SEPARATE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[Beth's 3]</td>
<td>[?]</td>
<td>[FROM Alex's 8]</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>[RE Beth's 3]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of concepts</th>
<th>1</th>
<th>2</th>
<th>4</th>
<th>2.33</th>
</tr>
</thead>
<tbody>
<tr>
<td># of inferences</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.00</td>
</tr>
</tbody>
</table>

8.15. Compare 3

Alex has 3 marbles. Bethany has 2 more marbles than Alex. How many marbles does Bethany have?

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>make set (Beth 2); referent found; infer JOIN; infer Beth also has an amount = to Alex's 3; infer Beth's total is result of action</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>[Alex's 3]</th>
<th>[JOIN Beth's 2]</th>
<th>[TO (Beth's 3)]</th>
<th>[RE Beth's total]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>[JOIN Beth's 2]</td>
<td>[TO (Beth's 3)]</td>
<td>[RE Beth's ?]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># of concepts</th>
<th>1</th>
<th>3</th>
<th>3</th>
<th>2.33</th>
</tr>
</thead>
<tbody>
<tr>
<td># of inferences</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.00</td>
</tr>
</tbody>
</table>
### 8.16. Compare 4

**Alex has 5 shells. Bethany has 2 shells less than Alex. How many shells does Bethany have?**

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>make set</td>
<td>make set (Beth has qty which is 2 less than Alex has) referent found; infer SEP-FROM; infer transformation of &quot;2 less 5&quot; to &quot;5 less 2&quot;; infer Beth's set is result of action</td>
<td>Beth ?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>[Alex's 5]</th>
<th>[SEPARATE 2]</th>
<th>[FR Alex's 5]</th>
<th>[RE Beth's ?]</th>
</tr>
</thead>
<tbody>
<tr>
<td># of concepts</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>2.33</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>1.00</td>
</tr>
</tbody>
</table>

### 8.17. Compare 5

**Alex has 8 marbles. Alex has 5 more marbles than Bethany. How many marbles does Bethany have?**

<table>
<thead>
<tr>
<th>Processing Summary</th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>make set</td>
<td>make set (Alex's 5); failed referent search counted as inference; infer Beth set for ref; infer JOIN; infer set of 8 is result of action</td>
<td>Bethany has ?</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Concepts in Memory</th>
<th>[Alex's 8]</th>
<th>JOIN</th>
<th>[Alex's 5]</th>
<th>[TO Beth's ?]</th>
<th>[RE Alex's 8]</th>
</tr>
</thead>
<tbody>
<tr>
<td># of concepts</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>3.00</td>
<td></td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>1.33</td>
<td></td>
</tr>
</tbody>
</table>
8.18. Compare 6

<table>
<thead>
<tr>
<th></th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Processing Summary</strong></td>
<td>make set (Alex has quantity that is 3 less than B’s); failed referent search counted as inference; infer SEPARATE-FROM; infer transformation from &quot;3 less B’s&quot; to &quot;B’s less 3&quot;; infer Alex’s 5 is the result of the action</td>
</tr>
<tr>
<td><strong>Concepts in Memory</strong></td>
<td>[Alex’s 5]</td>
</tr>
<tr>
<td></td>
<td>[FROM B’s ?]</td>
</tr>
<tr>
<td></td>
<td>[RE Alex’s 5]</td>
</tr>
<tr>
<td><strong># of concepts</strong></td>
<td>1</td>
</tr>
<tr>
<td><strong># of inferences</strong></td>
<td>0</td>
</tr>
</tbody>
</table>
APPENDIX C

Additional Network of Problems

This appendix continues the analysis of Chapter 5 by presenting each of the 18 traditional wordings in Riley et al.'s (1983, 1988) classification and the ways in which traditional wordings within each category can be altered in order to increase or decrease the probability of solution. For each problem in Riley et al.'s set of 18, new task characteristics are introduced, individually and/or in combination, to generate new wordings. Depending on the task characteristic introduced, each reworded version is then classified as easier or harder than the traditional wording. Where possible, empirical studies are cited which confirm a relative order of difficulty between two problems. In addition, the predictor equations as derived from the regression analyses in Chapter 4 are used to confirm the empirical results and/or predict the effect of each task characteristic on the probability of solution. The result is sets of four to six potential rewodorings in a relative order of difficulty for each of the original 18 benchmark problems.

9.1. Change Problems

Change problems are unique because they include action language in significant places, e.g., "gives" and "finds," as opposed to the more static language, e.g., "have altogether" and "have more than" in Combine and Compare problems. In the traditional Change problems of Riley et al. (1983), the action involves a transfer of possession of a set of objects between two people. Change problems vary in complexity by modifying the location of the unknown and the direction of the transfer, i.e., depending on whether a transferred amount is joined to an initial amount or is separated from an initial amount. The following sections present each of the six types of traditional Change problems and a number of ways to make those problems both easier and harder. A relative order of difficulty for some reworded versions have empirical backing while the order of others is predicted from the computer model.

9.1.1. Making Change 1 Problems Easier and Harder

Change 1 problems are presented in full in Chapter 5.

9.1.2. Making Change 2 Problems Easier

Figure 20 shows Riley et al.'s (1988) Change 2 problem. In Change 2 problems, a specific amount is separated from a previously owned amount with the result of the separation unknown. While most children can easily solve this problem (61% success rate for grade K and 100% success for all other grades), there are still at least two potential ways to alter the problem to make it easier: (i) include a related co-actor and (ii) include "mathematical" words which can facilitate the mathematical connections between sentences.

The first potential alternative to the traditional wording of Change 2 is to substitute a related co-actor (e.g., his sister) for the unrelated co-actor Bethany, as shown in Figure 21. The traditional wording

---

Equate problems also include action language in significant places, e.g., "need to get to have as many,". As explained in Chapter 5, equalizing situations are considered an easier form than the strict relational language found in Compare problems, thus Equate problems are discussed in the section on Compare problems.
Alex had 5 marbles. Then Alex gave Bethany 3 marbles. How many marbles does Alex have now?

Figure 20.
Traditional Change 2

Alex had 5 marbles. Then Alex gave his sister 3 marbles. How many marbles does Alex have now?

Figure 21.
Change 2 with related co-actor

of Change 2 in no way indicates Bethany as related to Alex and therefore does not provide any information which might lead to a more integrated representation of the situation. As discussed in Chapter 5, when the problem is worded with related co-actors, the protagonist, Alex, remains the center of focus during comprehension and the situation is more likely to be coordinated from his perspective.

The second alternative to the traditional wording of Change 2 is to include words which might facilitate text integration between sentences by highlighting the relationship between quantities. As discussed in Chapter 3 in the context of the computer model, the phrase of them generates an explicit reference to a previously mentioned set. In the case of Change 2, including the phrase of them in the second sentence, e.g.,

Then Alex gave Bethany 3 of them.

identifies that the three given to Bethany are part of the set of five he already had in his possession. When the model reads the traditional wording, an inference is required to establish that Bethany’s new set of three came from Alex’s original set of five, as opposed to three unrelated marbles. By including the phrase of them, the model predicts a potential increase in solution probability for children in grade K from 56% to 68%.

The word left can also facilitate text integration. Including the word left in the last sentence, e.g.,

How many does Alex have left?
can both reinforce the fact that a SEPARATE-FROM action is needed and that the amount in question is the result of that action. In the computer model, the word left reinforces the previously established SEPARATE-FROM arithmetic action but does not affect the number of inferences. Whether introducing the word left can actually help some children remains a working hypothesis.

9.1.3. Making Change 2 Problems Harder

The traditional Change 2 problem can be made more difficult by (i) constructing a more sparsely worded problem, (ii) making an inconsistent narrative focus or (iii) altering the presentational structure of the problem such that the text sequence does not match the natural order of events as they would happen in the real world.

In all Change problems, including Change 2, specific language is used to indicate the relational time order of events (e.g., then, after that) or to refer to specific points in time (e.g., in the beginning, now). If such language is removed, children must infer the order of events by relying on (i) the tense of sentences or (ii) the default strategy that sentences appear in chronological order. As discussed in Chapter 5 for Change 1 problems, there is no empirical evidence that the omission of these words would cause this problem to be more difficult. In short, the role of omitting relational time words is hypothesized to make Change 2 problems more difficult but remains an open question.

Since the traditional Change 2 problem is worded such that the narrative focus is consistent, that is, the protagonist Alex always remains in the grammatical subject position of every sentence, the second sentence of Change 2 can be reworded to produce an inconsistent narrative focus, e.g.,

Bethany got 3 marbles from Alex.

Although Staub and Reusser (1992) did not include Change 2 in their study, their general results indicate that such a change may be significant for all Change problems.

The final rewording change which makes Change 2 more difficult is to alter the time sequence of the problem such that the sentences are not in strict chronological order, as shown in Figure 22. A
Today Alex gave Bethany 3 marbles. Yesterday Alex had 5 marbles. How many marbles does Alex have now?

Figure 22.
Altering the time sequence of Change 2

critical difference between the traditional wording and this time-altered version is explained through the computer model when it solves these two problems similar to a novice who solves word problems in a sentence-by-sentence fashion. Table 35 summarizes the model's processing as it reads and solves the traditional version. Of particular interest is that the model makes the inference in the second sentence that the three marbles that Alex gives to Bethany are part of his original set of five. A SEPARATE-FROM action is instantiated in sentence two without an inference since the "transfer-out" situation is explicitly described in the problem text.

Table 35.
Summary of model's processing for traditional Change 2

<table>
<thead>
<tr>
<th>Alex had 5 marbles, Then Alex gave Bethany 3 marbles, How many marbles does Alex have now?</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Processing Summary</td>
<td>make set</td>
</tr>
<tr>
<td>Concepts in Memory</td>
<td>[Alex's 5]</td>
</tr>
<tr>
<td># of concepts</td>
<td>1</td>
</tr>
<tr>
<td># of inferences</td>
<td>0</td>
</tr>
</tbody>
</table>
In the time-altered version, the transfer of three to Bethany occurs in the first sentence. Since neither Alex nor Bethany have previous sets in working memory, the model does not interpret this as a transfer-out situation. Rather, the model simply creates a set of 3 for Bethany. The second sentence indicates that Alex had 5 yesterday and the model makes another set for Alex with this amount. Since the model is reading in a sentence-by-sentence fashion, it is not able to "rerun" the first two sentences over in a chronological fashion. After interpreting the third sentence to mean a request for the amount of marbles that Alex has now, the model notes that the three sets are differentiated by time and makes the following three inferences: (i) that the 3 today (those given to Bethany) are part-of yesterday’s 5 (Alex’s original amount); (ii) that Alex’s marbles now are also part-of his set of 5 marbles-yesterday; and (iii) that the 3 should be SEPARATED-FROM the 5. Table 36 summarizes the model’s processing as it reads and solves this time-altered version.

<table>
<thead>
<tr>
<th>Table 36, Summary of model’s processing for time-altered Change 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Today Alex gave Bethany 3 marbles. Yesterday Alex had 5 marbles.</strong></td>
</tr>
<tr>
<td><strong>How many marbles does Alex have now?</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Processing Summary</td>
</tr>
<tr>
<td>1. make set</td>
</tr>
<tr>
<td>2. make set</td>
</tr>
<tr>
<td>3. infer part-of; infer part-of; infer SEP-FROM action</td>
</tr>
<tr>
<td>Sentence</td>
</tr>
<tr>
<td>1. [today 3]</td>
</tr>
<tr>
<td>2. [today 3]</td>
</tr>
<tr>
<td>3. [SEP today 3]</td>
</tr>
<tr>
<td>4. [FR yest. 5]</td>
</tr>
<tr>
<td>5. [RE now ?]</td>
</tr>
<tr>
<td>Concepts in Memory</td>
</tr>
<tr>
<td>number of concepts</td>
</tr>
<tr>
<td>1. 2</td>
</tr>
<tr>
<td>2. 3</td>
</tr>
<tr>
<td>3. 2.00</td>
</tr>
<tr>
<td>number of inferences</td>
</tr>
<tr>
<td>0. 0</td>
</tr>
<tr>
<td>3. 3.00</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Using the predictor equations from the regression analyses, the model can help confirm that the time-altered version is indeed more difficult. Table 37 compares the model's predictions on the time-altered versions using first and third grade children's solution probabilities on problems of this type with the results from the Staub and Reusser (1992) study.

As with Change 1 time-altered versions, the model predicts that changing the order of events is difficult for young problem solvers. The model is conservative in its estimates of difficulty in relation to the observed performance of children in the Staub and Reusser study.

In summary, there are a number of potential features which can be altered to make the traditional Change 2 problem easier or harder. Words which facilitate the integration of text (e.g., *of them*) and the use of related co-actors may make this problem easier. Removing time relation language, introducing an inconsistent narrative focus and altering the time sequence may make this problem harder. Table 38 summarizes the five various wording types. The five problems are listed in an order of difficulty from easiest to most difficult as established by empirical studies and/or predicted by the computer model. As described in Chapter 5 for Change 1, a number of alternate rewordings are possible within each type which add to the total number of possible versions. Words in parentheses and bold, e.g., *(more, altogether)*, are words or phrases which can be inserted to facilitate the relation between sets. Words in

<table>
<thead>
<tr>
<th>Grade</th>
<th>Model's Prediction</th>
<th>Staub Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st Grade</td>
<td>0.58</td>
<td>0.31</td>
</tr>
<tr>
<td>3rd Grade</td>
<td>0.81</td>
<td>0.47</td>
</tr>
</tbody>
</table>
Table 38.
Change 2 (Result Unknown-SEPARATE-FROM) reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time;</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> Alex gave 3 of them to <em>(his sister/Bethany)</em>. How many <em>(soda cans)</em> does Alex have <em>(now, left)</em>?</td>
</tr>
<tr>
<td></td>
<td>Consistent Protagonist;</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Explicit Part</td>
<td></td>
</tr>
<tr>
<td>Sequential time;</td>
<td><em>(In the beginning)</em> Alex had 5 sodas cans. <em>(Then)</em> Alex gave 3 <em>(soda cans)</em> to <em>(his sister/Bethany)</em>. How many <em>(soda cans)</em> does Alex have <em>(now, left)</em>?</td>
<td></td>
</tr>
<tr>
<td>Consistent Protagonist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequential time;</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> <em>(his sister/Bethany)</em> got 3 <em>(soda cans)</em> from Alex. How many <em>(soda cans)</em> does Alex have <em>(now, left)</em>?</td>
<td></td>
</tr>
<tr>
<td>Inconsistent Protagonist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NonSequential time;</td>
<td>Today Alex gave 3 soda cans to <em>(his sister/Bethany)</em>. Yesterday Alex had 5 <em>(soda cans)</em>. How many <em>(soda cans)</em> does Alex have <em>(now, left)</em>?</td>
<td></td>
</tr>
<tr>
<td>Consistent Protagonist</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time;</td>
<td>Today <em>(Alex's sister/Bethany)</em> got 3 soda cans from Alex. Yesterday Alex had 5 <em>(soda cans)</em>. How many <em>(soda cans)</em> does Alex have <em>(now, left)</em>?</td>
</tr>
<tr>
<td></td>
<td>Inconsistent Protagonist</td>
<td></td>
</tr>
</tbody>
</table>

Parentheses and italized, e.g., *(In the beginning, now)*, are words or phrases which can be inserted to increase the explicitness of time relations. Words in brackets, e.g., *(soda cans)*, are optional ellipsed phrases. For example, after the initial mention of soda cans, the phrase may be removed from subsequent sentences or just the word “cans” may be used. Words or phrases in brackets, e.g., *(his sister/Bethany)*, refer to the option of *(related/unrelated) coactors. In general, it is hypothesized that problems which include time and set relation language and which involve related coactors can increase solution probability within each of the four versions.
9.2. Making Change 3 Problems Easier

Figure 23 shows Riley et al.'s (1988) Change 3 problem. In Change 3 problems, an unknown amount is joined to a previously owned amount. The resulting amount of the join is then given, leaving the amount of the transfer-in unknown. While many children can solve this problem (22% success for grade K, 61% for first grade, 80% for second and 95% for third), there are still at least four potential ways to alter the problem to make it easier: (i) include a related co-actor; (ii) use a consistent narrative focus; (iii) include "mathematical" words which can highlight the superset; and (iv) reword the problem such that the question suggests a direct modeling strategy.

The rationale for the first and second suggestions, that is, including related co-actors and a consistent narrative focus, are discussed for Change 1 and 2. In the results of Staub and Reusser (1992), the presence of related co-actors was a more dominant factor than consistent narrative focus. In fact, narrative focus was only a factor when the co-actors were not related.

A third potential alternative to the traditional wording of Change 3 is to include words which might facilitate text integration between sentences, e.g., by noting the uniqueness of sets or highlighting which set is the superset.

As discussed in Chapter 3, including the word *more* in the second sentence, e.g., "Then Bethany gave Alex some *more* marbles," highlights that the unknown amount given to Alex is separate from those he previously had. The "in addition to" lexical entry for the word "more" as used in this example

| Alex had 5 marbles. Then Bethany gave Alex some marbles. Now Alex has 8 marbles. How many marbles did Bethany give him? |

---

**Figure 23.**

Traditional Change 3
(that is, Bethany gave Alex some in addition to the ones he already had) is also used by the model to interpret relational phrases such as “how many more than” in a procedural in-addition-to fashion.

The word altogether causes the model to partition a set along some dimension (e.g., by ownership or color). In the case of Change 3, including the word altogether in the third sentence, e.g.,

Now Alex has 8 marbles altogether.

highlights that Alex’s marbles can be partitioned into two subsets: the original amount and the transferred amount of some. The results of Davis-Dorsey et al. (1991) show that the inclusion of the word altogether can have a positive influence on second grade children’s comprehension. The computer model, on the other hand, does not predict that including altogether in this problem will increase solution probability. In the simulation, the word now at the beginning of the third sentence marks that set as the result of the previous transfer into the original set and thus the superset. However, since altogether implies a JOIN situation in the model, its use marks the second instance of a need to JOIN (the first being from the explicit transfer-in action) so the already instantiated JOIN action is reinforced. Although reinforcement is not included in the model’s predictor variables, its presence in conjunction with the Davis-Dorsey et al. (1991) results indicate that including altogether may help some children.

The last potential version that can make the traditional Change 3 easier is to word the question such that it implies a direct modeling “add-on” strategy. Figure 24 shows such a problem as used in the Carpenter and Moser (1983) studies. There is some debate in the literature whether a Change 3 problem is at a level of difficulty similar to Change 1 and 2 or if a Change 3 problem is more difficult than these two easiest Change problems. Both Briars and Larkin (1984) and Riley et al. (1983) hypothesize that Change 1 and 2 can be solved prior to Change 3 since the ability to solve missing-addend problems like Change 3

---

1 Davis-Dorsey et al. (1991) did not publish solution probabilities. Rather, children were scored on a zero to two point system, where they were awarded one point for an “appropriate problem representation” and another point if they gave the correct answer. On standard word problems, that is when problems did not contain context personalization affects (e.g., the names of friends, the name of a favorite movie), the study found a nonsignificant increase from 0.43 to 0.51 in second graders, although it is in the predicted direction. However, when context personalization versions were used, the inclusion of the word altogether significantly increased the scores from 0.31 to 0.77. As noted in their study, increases in comprehension were most pronounced when rewording was associated with context personalization.
Alex had 5 marbles. How many marbles does Alex need to get to have 8 altogether?

Figure 24.
"Need to Get" version of Change 3

emerges at the same time as the more complex add-on counting strategy. In the Riley et al. (1983) model, the traditional Change 3 is hypothesized to be more difficult than Change 1 and 2 and this hypothesis is confirmed in their studies with children. The study of Carpenter and Moser (1983), however, used the easier version of Change 3 and found higher levels of performance than in Riley's results with the traditional Change 3.

Thus, if there is a stage in which children can solve addition problems (Change 1) before they can solve missing-addend problems, most of the children in the study must have passed through this stage before they entered first grade (Carpenter & Moser, 1984, p. 192).

The computer model also predicts that the Carpenter and Moser (1983) version is more similar in difficulty to Change 1 and 2 than the Riley (1983) version. Table 39 summarizes the empirical results of these two versions and the associated predictions from the model for second grade students.

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Model's Prediction</th>
<th>Carpenter &amp; Moser Data</th>
<th>Riley Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>0.81</td>
<td>-</td>
<td>0.80</td>
</tr>
<tr>
<td>&quot;Need to Get&quot;</td>
<td>0.96</td>
<td>0.93</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 39.
Comparison of Riley and Carpenter versions of Change 3 and predictions for second grade
9.3. Making Change 3 problems harder

The traditional Change 3 problem can be made potentially more difficult by (i) constructing a more sparsely worded problem by removing the time-relational language (e.g., *then*) and/or (ii) altering the presentational structure of the problem such that the text sequence does not match the natural order of events as they would happen in the real world. Each of these rewordings and their corresponding features which make the problem more difficult are described in Change 1 and 2.

In summary, there are a number of potential features which can be altered to make the traditional Change 3 problem easier or harder. Words which facilitate the integration of text (e.g., *more, altogether*), the use of related co-actors, consistent narrative focus and wording that highlights a direct modeling strategy can make this problem easier. Removing the time relation language and altering the time sequence can make this problem harder. Table 40 summarizes the five various wording types. The five problems are listed in an order of difficulty from easiest to most difficult as established by empirical studies and/or predicted by the computer model. As described above, a number of alternate rewordings are possible within each type which add to the number of total possible versions.

9.4. Making Change 4 Problems Easier

Figure 25 shows Riley et al.'s (1988) Change 4 problem. In Change 4 problems, an unknown amount is separated from a previously known amount. The resulting amount of the separation is then given. The success rates for this problem across the early grades are: 30% success for grade K; 61% for first, and 100% success for grades two and three. Although the older children do not have a problem with the traditional wording of Change 4, there are at least two potential ways to alter the problem to make it easier for children in grades K and 1: (i) include a related co-actor and (ii) include "mathematical" words which can facilitate text integration between sentences.

The first potential alternative to the traditional wording of Change 4 is to substitute a related co-actor (e.g., *his sister*) for the unrelated co-actor Bethany, as discussed above. The second alternative is to
<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time; Consistent Protagonist; Direct Modeling Hint</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. How many <em>(more)</em> <em>(soda cans)</em> does Alex need to get <em>(from his sister/Bethany)</em> to have 8 <em>(altogether)</em>?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; Consistent Protagonist</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> Alex got some <em>(more)</em> <em>(soda cans)</em> from <em>(his sister/Bethany)</em>. <em>(Now)</em> Alex has 8 <em>(soda cans)</em> <em>(altogether)</em>. How many <em>(more)</em> <em>(soda cans)</em> did Alex get from <em>(his sister/Bethany)</em>?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; InConsistent Protagonist</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> <em>(his sister/Bethany)</em> gave Alex some <em>(more)</em> <em>(soda cans)</em>. <em>(Now)</em> Alex has 8 <em>(soda cans)</em> <em>(altogether)</em>. How many <em>(more)</em> <em>(soda cans)</em> did <em>(his sister/Bethany)</em> give Alex?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; Consistent Protagonist</td>
<td>Today <em>(Alex’s sister/Bethany)</em> gave some soda cans to Alex. Yesterday <em>(Alex)</em> had 5 <em>(soda cans)</em>. Now <em>(Alex)</em> has 8 <em>(soda cans)</em> <em>(altogether)</em>. How many <em>(more)</em> <em>(soda cans)</em> did <em>(his sister/Bethany)</em> give to <em>(Alex)</em>?</td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time; InConsistent Protagonist</td>
<td>Today <em>(Alex’s sister/Bethany)</em> gave some soda cans to <em>(Alex)</em>. Yesterday <em>(Alex)</em> had 5 <em>(soda cans)</em>. Now <em>(Alex)</em> has 8 <em>(soda cans)</em> <em>(altogether)</em>. How many <em>(more)</em> <em>(soda cans)</em> did <em>(his sister/Bethany)</em> give to <em>(Alex)</em>?</td>
</tr>
</tbody>
</table>
Alex had 5 marbles. Then Alex gave Bethany some marbles. Now Alex has 2 marbles. How many marbles did Alex give Bethany?

Figure 25.
Traditional Change 4

include words which might facilitate the links between sentences by highlighting the relationship between quantities. As discussed above for Change 2, the phrase of them generates an explicit reference to a previously mentioned set. By including the phrase of them, the computer model predicts a potential increase in solution probability for children in grade K from 22% to 31% and from 46% to 58% for children in first grade.

The word left can also facilitate text integration. Including the word left in the third sentence, e.g.,

Now Alex has 2 marbles left.

can both reinforce the fact that a SEPARATE-FROM action is needed and that Alex’s 2 marbles (now) are the result of that action. Like the proposed addition of related co-actors, introducing left to increase solution probability has no empirical backing and remains a working hypothesis.

9.5. Making Change 4 Problems Harder

The traditional Change 4 problem can be made more difficult by (i) constructing a more sparsely worded problem, (ii) making an inconsistent narrative focus or (iii) altering the presentational structure of the problem such that the text sequence does not match the natural order of events as they would happen in the real world. Each of these versions and their impact on solution probability is discussed more fully in Change 2 above.

In summary, there are a number of potential features which can be altered to make the traditional Change 4 problem easier or harder. Words which facilitate the integration of text (e.g., of them, left) and
the use of related co-actors can make this problem easier. Removing the time relation language, introducing an inconsistent narrative focus and altering the time sequence can make this problem harder. Table 41 summarizes the five various wordings. The five problems are listed in an order of difficulty from easiest to most difficult as established by empirical studies and/or predicted by the computer model.

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time; Consistent Protagonist; Explicit Part</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> Alex gave some of them to [his sister/Bethany]. <em>(Now)</em> Alex has 2 [(soda) cans] <em>(left)</em>. How many [(soda) cans] did Alex give to [his sister/Bethany]?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; Consistent Protagonist</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> Alex gave some [(soda) cans] to [his sister/Bethany]. <em>(Now)</em> Alex has 2 [(soda) cans] <em>(left)</em>. How many [(soda) cans] did Alex give to [his sister/Bethany]?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; InConsistent Protagonist</td>
<td><em>(In the beginning)</em> Alex had 5 soda cans. <em>(Then)</em> [his sister/Bethany] got some [(soda) cans] from Alex. <em>(Now)</em> Alex has 2 [(soda) cans] <em>(left)</em>. How many [(soda) cans] did [his sister/Bethany] get from Alex?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; Consistent Protagonist</td>
<td>Today [Alex’s sister/Bethany] got some soda cans from Alex. Yesterday Alex had 5 [(soda) cans]. <em>(Now)</em> Alex has 2 [(soda) cans] <em>(left)</em>. How many [(soda) cans] did [his sister/Bethany] get from Alex?</td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time; InConsistent Protagonist</td>
<td>Today [Alex’s sister/Bethany] got some soda cans from Alex. Yesterday Alex had 5 [(soda) cans]. <em>(Now)</em> Alex has 2 [(soda) cans] <em>(left)</em>. How many [(soda) cans] did [his sister/Bethany] get from Alex?</td>
</tr>
</tbody>
</table>
9.6. Making Change 5 Problems Easier

Figure 26 shows Riley et al.'s (1988) Change 5 problem. In Change 5 problems, a known amount is joined to a previously mentioned unknown amount. The resulting amount of the join is then given. This problem is significantly more difficult for children than the previous four Change problems. In general, it is not until the third grade that most children are able to handle the difficulty caused by the unknown being in the initial sentence. Again, using the Riley et al. (1988) data, solution probabilities for this problem are: 9% for kindergarten, 33% for first, 75% for second and 95% for third grade. There are three potential ways to alter the problem to make it easier: (i) include a related co-actor; (ii) use a consistent narrative focus and (iii) include "mathematical" words and relational phrases to highlight the distinction of and relation between the sets.

The rationale for the first and second suggestions, that is, including related co-actors and a consistent narrative focus, are discussed above in Change 1 and 2. The third potential rewording is to include the relational phrase "In the beginning" in the first sentence and the words "more" and "altogether" in the second and third sentences, as shown below in Figure 27. A common theme which emerges from previous studies with the traditional Change 5 problem is that children do not recognize or remember that Alex's initial group of some is indeed a set. For example, Cummins et al. (1988) note that one of children's most common errors with the traditional Change 5 wording is to give the answer "3," that is, the first specific amount that Alex had. Since Alex's some in the first sentence is not considered a set

---

Alex had some marbles. Then Bethany gave Alex 3 marbles. Now Alex has 8 marbles. How many marbles did Alex have in the beginning?

Figure 26.
Traditional Change 5
In the beginning Alex had some marbles. Then Bethany gave Alex 3 more marbles. Now Alex has 8 marbles altogether. How many marbles did Alex have in the beginning?

Figure 27.
Highlighting the uniqueness of the initial set in Change 5

(either because they do not yet understand the word some in a quantifier sense or because they are not able to remember that Alex had an unknown amount prior to getting 3 from Bethany), the three from Bethany are the amount of marbles he had "in the beginning."

Each of the rewordings, In the beginning, more and altogether, are an attempt to highlight the existence of Alex's initial set of marbles. In the beginning explicitly states that Alex's some existed prior to any other event in the situation to follow. The relational time phrase is also repeated in the question, which may reinforce the fact that this initial set is the one of interest.

Including the word more in the second sentence generates an implicit reference to the previously mentioned initial set, i.e., "more (than Alex's some)." This reference implies that the three given to Alex are independent of and in addition to those he already had in his possession. In short, more can highlight the existence of the initial set by highlighting that the three given to him are not the first marbles he had.

Including the word altogether in the third sentence can help children recognize Alex's set of 8 as the superset.³ Altogether generates a situation where the current set is partitioned along some dimension, in this case, time. That is, the 8 now are made up of those Alex just got from Bethany and those he previously had.

³ Of course this assumes that children understand the meaning of altogether. As Cummins (1991) has shown, this is often not the case. Young children may interpret altogether as each, which of course leads to quite different interpretations of what the problem is asking. At this point I am assuming that the classroom teacher or learning environment has already established that a child understands the word altogether and therefore that its inclusion in the problem is a potential help.
9.7. Making Change 5 Problems Harder

The traditional Change 5 problem can be made potentially more difficult by (i) constructing a more sparsely worded problem by removing the relational language (e.g., then), (ii) altering the presentational structure of the problem such that the text sequence does not match the natural order of events as they would happen in the real world, (iii) removing the initial sentence describing the unknown set and (iv) combining (ii) and (iii). The first two rewordings involving the use of relational language and the unnatural order of events are described above in Change 1 and 2.

The third suggestion involves removing the initial sentence of the traditional Change 5 problem, as shown below in Figure 28. From the above discussion on the difficulty children have with dealing with an initial set with an unknown amount, it is not surprising that researchers have found this reworded version to be more difficult (Davis-Dorsey et al., 1991; DeCorte, Verschaffel et al., 1985). For example, DeCorte found that 33% of first graders could solve the traditional Change 5 but only 13% could solve the reworded version. Similar reductions (79% traditional, 61% reworded) were found for second graders in that study. Table 42 summarizes children's differences in solution probability when they solve the traditional Change 5 and DeCorte's reworded version as well as the computer model's predictions on these two versions.

A final version which can make the traditional Change 5 more difficult is to combine the reordering of events (suggested by Staub and Reusser, 1992) with a version which is missing the sentence describing

Bethany gave Alex 3 marbles. Now Alex has 8 marbles. How many marbles did Alex have in the beginning?

Figure 28.
Traditional Change 5 with the initial sentence removed
Table 42.
Comparison of traditional and DeCorte versions of Change 5

<table>
<thead>
<tr>
<th>Problem Wording</th>
<th>Model’s Prediction</th>
<th>DeCorte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional (1st grade)</td>
<td>0.58</td>
<td>0.33</td>
</tr>
<tr>
<td>Missing Sentence (1st grade)</td>
<td>0.27</td>
<td>0.13</td>
</tr>
<tr>
<td>Traditional (3rd grade)</td>
<td>0.81</td>
<td>0.79</td>
</tr>
<tr>
<td>Missing Sentence (3rd grade)</td>
<td>0.62</td>
<td>0.61</td>
</tr>
</tbody>
</table>

the initial set (suggested by DeCorte, Verschaffel et al., 1985). Figure 29 gives an example of this combination. In this problem, events are not presented in their natural order of occurrence and no mention (prior to the question) is made that “Alex had some marbles yesterday.” Because each of these features in isolation can make the traditional Change 5 more difficult, one would assume that this combination would be very difficult. In fact, Staub and Reusser (1992) presented this difficult version to children and found this to be the case. 33% of first graders in the Riley et al. (1988) study were able to solve the traditional wording while only 10% of first graders in Staub’s study had success with the reworded version.

The computer model predicts that only 11% of first graders could solve the difficult Staub version, a very close match to the observed results. For third grade, 95% of the students in the Riley study were able to solve the traditional wording while only 15% of Staub’s third graders could solve the difficult version.

---

Alex has 8 marbles now. Today Alex got 3 marbles from Bethany. How many marbles did Alex have yesterday?

Figure 29.
Non-sequential time and missing initial set version of Change 5
The computer model, however, predicts that 86% of third graders could solve the difficult version. Clearly, Staub's reworded version requires comprehension processes that are not fully captured in the model's predictor equations for third graders. One explanation is that the inferences related to reorganizing the order of events are relatively complex compared to the inferences found in the benchmark set of eighteen problems in the regression analyses. The relatively close match between Staub's results and the model's predictions for first graders appears to indicate that young readers are very sensitive to slight increases in the number of inferences while third graders are unaffected by an additional inference, unless that inference is a relatively complex one.

In summary, there are a number of potential features which can be altered to make the traditional Change 5 problem easier or harder. Words which facilitate the integration of text (e.g., more, altogether), the use of related co-actors and consistent narrative focus can make this problem easier. Removing the time relation language, altering the time sequence, removing the sentence describing the initial set and a combination of non-sequential presentation and absence of the initial sentence can make this problem harder. Table 43 summarizes the five various wording types. The five problems are listed in an order of difficulty from easiest to most difficult.

9.8. Making Change 6 Problems Easier

Figure 30 shows Riley et al.'s (1988) Change 6 problem. In Change 6 problems, a known amount is separated from an unknown amount. The resulting amount of the separation is then given. Like Change 5, this is a relatively complex Change problem. The success rates for this problem across the early grades are: 17% success for grade K, 39% for first, 65% for second and 90% for third. There are at least two potential ways to alter the traditional wording of Change 6 to make it easier for children to comprehend: (i) include a related co-actor and (ii) include "mathematical" words which can facilitate text integration between sentences. As discussed above for Change 2 and 4, including the phrase of them in the second sentence, e.g., "Then Alex gave Bethany 3 of them," generates an explicit reference to
Table 43.
Change 5 (Initial Unknown-JOIN) Reworded Versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time; Consistent Protagonist</td>
<td>(In the beginning) Alex had some soda cans. (Then) Alex got 3 (more) [(soda) cans] from [his sister/Bethany]. (Now) Alex has 8 [(soda) cans] (altogether). How many [(soda) cans] did Alex have in the beginning?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; InConsistent Protagonist</td>
<td>(In the beginning) Alex had some soda cans. (Then) [his sister/Bethany] gave Alex 3 (more) [(soda) cans]. (Now) Alex has 8 [(soda) cans] (altogether). How many [(soda) cans] did Alex have in the beginning?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; Consistent Protagonist</td>
<td>Today Alex got 3 soda cans from [his sister/Bethany]. Yesterday Alex had some [(soda) cans]. Now Alex has 8 [(soda) cans] (altogether). How many [(soda) cans] did Alex have yesterday?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; InConsistent Protagonist</td>
<td>Today [Alex’s sister/Bethany] gave 3 soda cans to Alex. Yesterday Alex had some [(soda) cans]. Now Alex has 8 [(soda) cans] (altogether). How many [(soda) cans] did Alex have yesterday?</td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time; InConsistent Protagonist; Missing Initial State</td>
<td>Today [Alex’s sister/Bethany] gave 3 soda cans to Alex. Now Alex has 8 [(soda) cans] (altogether). How many [(soda) cans] did Alex have yesterday?</td>
</tr>
</tbody>
</table>

Alex had some marbles. Then Alex gave Bethany 3 marbles. Now Alex has 2 marbles. How many marbles did Alex have in the beginning?

Figure 30.
Traditional Change 6

Alex’s previously mentioned set of some marbles. When this phrase is included, the computer model
predicts a potential increase in solution probability for children in all four grades levels.

Introducing the word *left* in the third sentence, e.g., “Now Alex has 2 marbles *left*,” can also facilitate text integration by reinforcing the fact that SEPARATING three FROM Alex’s initial set results in a set of two.

9.9. Making Change 6 Problems Harder

The traditional Change 6 problem can be made potentially more difficult by (i) removing the relational language, (ii) making an inconsistent narrative focus, (iii) altering the presentational structure of the problem such that the text sequence does not match the natural order of events as they would happen in the real world, (iv) removing the sentence describing the initial set or (v) a combination of non-sequential presentation and absence of the initial set. Each of these versions and its impact on solution probability is discussed more fully in Change 2, 4 and 5 above.

In summary, there are a number of potential features which can be altered to make the traditional Change 6 problem easier or harder. Words which facilitate the integration of text (e.g., *of them, left*) and the use of related co-actors can make this problem easier. Removing the time relation language, introducing an inconsistent narrative focus, altering the time sequence, or removing the sentence describing the unknown initial set can make this problem harder. Table 44 summarizes the five various wording types.

10. Combine Problems

Combine problems involve the relationships between a particular set and its two disjoint subsets. The following sections present each of the six types of traditional Combine problems and a number of ways to make those problems both easier and harder. A relative order of difficulty for some reworded versions have empirical backing while the order of others is predicted from the computer model.
### Table 44.
Change 6 (Initial Unknown-SEPARATE-FROM) reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Sequential time; Consistent Protagonist; Explicit Part</td>
<td>*(In the beginning) Alex had some soda cans. <em>(Then)</em> Alex gave 3 of them to {his sister/Bethany}. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have in the beginning?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; Consistent Protagonist</td>
<td>*(In the beginning) Alex had some soda cans. <em>(Then)</em> Alex gave 3 {soda} cans to {his sister/Bethany}. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have in the beginning?</td>
</tr>
<tr>
<td></td>
<td>Sequential time; InConsistent Protagonist</td>
<td>*(In the beginning) Alex had some soda cans. <em>(Then)</em> {his sister, Bethany} got 3 {soda} cans from Alex. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have yesterday?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; Consistent Protagonist</td>
<td>Today Alex gave 3 soda cans to {his sister/Bethany}. Yesterday Alex had some {soda} cans. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have yesterday?</td>
</tr>
<tr>
<td></td>
<td>NonSequential time; InConsistent Protagonist</td>
<td>Today {Alex's sister/Bethany} got 3 soda cans from Alex. Yesterday Alex had some {soda} cans. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have yesterday?</td>
</tr>
<tr>
<td>hardest</td>
<td>NonSequential time; InConsistent Protagonist; Missing Initial Set</td>
<td>Today {Alex's sister/Bethany} got 3 soda cans from Alex. <em>(Now)</em> Alex has 2 {soda} cans {left}. How many {soda} cans did Alex have yesterday?</td>
</tr>
</tbody>
</table>

### 10.1. Making Combine 1 and 2 Problems Easier

Figure 31 shows Riley et al.'s (1988) Combine 1 problem. In Combine 1 and 2 problems, two disjoint sets (e.g., two sets with different owners) are described and the question requests the amount of a third set which is the result of joining or combining the two disjoint sets. Of particular importance in

---

4 Combine 2 problems are exactly the same as Combine 1 except the following sentence is added to the beginning: “David and Kathy had some soda cans.” As shown in the Riley et al. (1988) results, adding this sentence at the beginning of the problem has little effect on children’s comprehension, thus these two problems are considered together in this section.
David has 5 soda cans. Kathy has 3 soda cans. How many soda cans do they have altogether?

Figure 31.
Traditional Combine 1

these two types of problems is that the act of \textit{joining} the two disjoint sets into a third (super)set is not explicitly stated in the text. Unlike Change problems which involve significant action language, such as \textit{giving}, Combine problems use static language, such as \textit{altogether}, to describe the result of the act of joining two sets together. As shown earlier, Combine problems are in general harder than Change problems. Briars and Larkin (1984) and others argue that this difference is due largely to the absence of action language in Combine problems.

While almost all children can solve Combine 1 problems (74\% in grade K, 100\% for all other grades) and Combine 2 problems (70\% in grade K, 100\% for all other grades), there are three potential rewordings which might help those few students who are not able to solve this type of problem: (i) using related co-actors, (ii) repeating the specific names of the owners in the question rather than using the pronoun, “they,” and (iii) using action language to explicitly mention the \textit{join} action.

The first suggestion, the use of related co-actors (e.g., David and \textit{his mother}), stems from the empirical work of Staub and Reusser (1992) as discussed earlier. There is currently no empirical support that this change in and of itself would help make Combine problems in general easier to solve. Unlike Change problems where it is beneficial to keep a focus on the actor of interest, Combine problems are not “stories,” and thus neither of the actors is more important in this problem. In short, the use of related co-actors to make Combine 1 and 2 problems easier is a working hypothesis with weak support.

The second potential rewording is to repeat the owners names in the question rather than using the pronoun \textit{they}, e.g.,
How many soda cans do David and Kathy have altogether?

While the use of pronouns has been suggested as a potential source of difficulty in problem comprehension (e.g., Verschaffel et al., 1992), there is no clear evidence that children have difficulty with pronouns in arithmetic word problems (cf. Stern, 1993).

The third potential rewording is to include language which describes a hypothetical situation where the two disjoint subsets are joined, as shown in Figure 32. While this version has not been tested, it is clear from previous results that action language has a profound effect on children's success in general (Briars & Larkin, 1984) and specifically on other types of problems (Carpenter & Moser, 1981, 1983; Hudson, 1983; Stern and Lehndorfer, 1992). Because the join action is explicitly mentioned in this version, no inference is needed to infer the join action in the computer model. The computer model predicts that this newly worded version will be easier than the traditional wording for children in kindergarten (68% predicted success rate for the new version as compared to a prediction of 56% for the traditional wording).

10.2. Making Combine 1 and 2 Problems Harder

Combine 1 and 2 problems can be made more difficult by: (i) removing the word altogether or (ii) distinguishing subsets by abstract relations, such as time or actions, rather than by ownership or agent.

In the computer model, the word altogether implies that the current set is partitioned according to some dimension such as ownership or color and that two subsets which vary along this dimension can be

David has 5 soda cans. Kathy has 3 soda cans. If David and Kathy put their soda cans together, how many soda cans will they have altogether?

Figure 32.
Action language in Combine 1
JOINed to form the current set. When the word *altogether* is not in the traditional wording, e.g.,

How many do they have?

there is no implication that the current set is partitioned. The model must infer the partition based on
other factors, such as the word *they* which references two previous owners. As discussed above, an
increase in the amount of inferencing is predicted to make the problem more difficult.

Distinguishing subsets by abstract relations rather than by physical qualities such as owner or color
may also cause the traditional wording to be more difficult. In a traditional wording, the two subsets may
vary by ownership, also referred to as varying by *agents* (Nesher, 1982). Other *qualifying dependencies* in
the Nesher (1982) classification include *adjectives* (e.g., big, small, big and small), *arguments* (e.g., boys,
girls, children), *location* (e.g., on the table, on the shelf, in the room) and *time* (e.g., yesterday, today,
both these days).\(^5\) While the computer model is not currently sensitive to changes in the type of qualifying
dependency, the hypothesis is that the level of abstraction in the units to be counted can affect solution probability. In the traditional wording, discrete objects (e.g., marbles, soda cans) are counted and added. It is also possible to have children count more abstract quantities, such as miles or "the number of times an act is performed." For example, Figure 33 shows a problem which requires that "times" be
treated as an abstract entity in order to count "3 times" and "2 times."

In summary, Combine 1 and 2 problems can be modified to be potentially easier or harder than the

Patrick ran to the store 3 times. Kevin ran to the store 2 times. How many times did they run
altogether?

**Figure 33.**

Subsets in a Combine 1 problem differentiated by the abstract "number of times"
Table 45.
Combine 1 and 2 (Combination Unknown-JOIN) reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Hypothetical Action</td>
<td>David has 5 soda cans.  {Kathy/His mother} has 3 [(soda) cans]. If (David and {Kathy/His mother}, they) put them together, how many [(soda) cans] will (David and {Kathy/His mother}, they) have (altogether)?</td>
</tr>
<tr>
<td></td>
<td>Traditional Wording</td>
<td>David has 5 soda cans.  {Kathy/His mother} has 3 [(soda) cans]. How many [(soda) cans] do (David and {Kathy/His mother}, they) have (altogether)?</td>
</tr>
<tr>
<td>hardest</td>
<td>Abstract Differentiation</td>
<td>David ran to the store 3 times.  {Kathy/His mother} ran to the store 2 times. How many times did they run (altogether)?</td>
</tr>
</tbody>
</table>

traditional wording. A hypothesized ordering of relative difficulty is given in Table 45.

10.3. Making Combine 3 and 4 Problems Easier

Figure 34 shows Riley et al.’s (1988) Combine 3 problem. In Combine 3 and 4 problems, two disjoint subsets, one with a known amount and the other with an unknown amount, are initially described.

David has 5 soda cans. Kathy has some soda cans. They have 8 soda cans altogether. How many soda cans does Kathy have?

Figure 34.
Traditional Combine 3

\footnote{In Nesher’s results, the differences in children’s solution probabilities on problems with these qualifiers (agent, adjective, argument, location and time) were significant only on those problems which involved subtraction. Subtraction problems involving subsets which varied by argument and location were found to be slightly easier than those problems which varied by agent.}
The superset of these sets is then introduced with a known amount. The question requests the amount of the unknown subset.6

There are three potential rewordings which might help those students who are not able to solve this type of problem: (i) the use of related co-actors (e.g., David and his mother), (ii) repeat the specific names of the owners describing the superset rather than using a pronoun (e.g., David and Kathy rather than they), and (iii) the use of action language to explicitly mention the join action (e.g., describe the union of the two disjoint sets with a hypothetical situation involving action language: “If David and Kathy put their soda cans together, they will have 8 altogether”). As discussed for Combine 1 and 2, there is currently no empirical support that these rewordings can help children. In addition, the current computer model is not sensitive to the first two alternatives. The model does predict, however, that the third alternative involving the use of action language will be easier than the traditional wording since the model does not have to infer the Join action.

10.4. Making Combine 3 and 4 Problems Harder

There exist two potential rewordings which can make the traditional Combine 3 and 4 problems harder: (i) removing the word altogether and (ii) increasing the level of abstraction of the semantic relation between the disjoint sets. Both of these alternatives are discussed more fully in Combine 1 and 2 above.

Table 46 summarizes the hypothesized ordering of problem rewordings relative to the traditional wordings of Combine 3 and 4.

---

6 Combine 3 and 4 differ only in the order of the first two sentences. In Combine 3, the second sentence contains the unknown set whereas in Combine 4 the unknown set is introduced in the first sentence. While the Raley et al. (1988) results do show some differences in solution probability between these two problems (Combine 3: 4%-K, 39%-1st, 85%-2nd, 100%-3rd and Combine 4: 13%-K, 39%-1st, 70%-2nd, 100%-3rd), the differences are slight. In addition, the computer model predicts that both of these problems are equally difficult, thus they are considered together in this section.
Table 46.
Combine 3 and 4 (Subset Unknown-JOIN) reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>Hypothetical Action</td>
<td>David has 5 soda cans. [Kathy/His mother] has some [(soda) cans]. If (David and (Kathy/ his mother), they) put them together, they will have 8 [(soda) cans] (altogether). How many [(soda) cans] does [Kathy/ his mother] have?</td>
</tr>
<tr>
<td></td>
<td>Traditional Wording</td>
<td>David has 5 soda cans. [Kathy/His mother] has some [(soda) cans]. They have 8 [(soda) cans] (altogether). How many [(soda) cans] does [Kathy/ his mother] have?</td>
</tr>
<tr>
<td>hardest</td>
<td>Abstract Differentiation</td>
<td>David ran to the store 5 times. [Kathy/His mother] ran to the store some number of times. They ran to the store 8 times (altogether). How many times did [Kathy/ his mother] run to the store?</td>
</tr>
</tbody>
</table>

10.5. Making Combine 5 and 6 Problems Easier and Harder

Combine 5 and 6 problems are presented in full in Chapter 5.

11. Compare Problems

Compare problems involve a comparison between two distinct, disjoint subsets. The following sections present each of the six types of traditional Compare problems and a number of ways to make those relatively complex problems easier. A relative order of difficulty for some reworded versions have empirical backing while the order of others is predicted from the computer model.

11.1. Making Compare 1 Problems Easier

Figure 35 shows Riley et al.'s (1988) Compare 1 problem. In Compare 1 problems, the cardinalities of two disjoint sets are compared. The question requests the difference between the sets with a focus
Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans does Chrissy have more than Jacob?

Figure 35.
Traditional Compare 1

on the larger set, i.e., "how many more?" Compare problems are in general the most difficult of the three types of semantic structure and this problem in particular is solved by only an average of 53% of the children in the grades K through 3. Three problem rewordings can potentially make this problem easier than the traditional wording. The first two rewordings have empirical backing: (i) replace the static relational language (how many more than) with action language which facilitates a one-to-one correspondence matching strategy and (ii) replace the static relational language (how many more than) with action language which describes a hypothetical act of making the two sets equal (e.g., how many does Jacob need to get to have as many as Chrissy). The third alternative involves the use of (iii) related co-actors.

The use of related co-actors (e.g., Jacob and his cousin) has been discussed above. While the relationship between the sets is clearly an important element in these types of problems, the impact of using related co-actors remains but a working hypothesis.

Replacing the static relational language (how many more than) with action language (e.g., how many won’t get to describe the difference set) is the first suggestion for making the traditional wording of Compare 1 easier. In a study by Hudson (1983), young children were presented with "won’t get" versions of the Compare 1 problem of the type shown in Figure 36. The study produced the result that almost all children in kindergarten and first grade could solve the "Won’t Get" version, despite being unable to solve the traditional "more than" version. According to the categorization of semantic structure found in Riley and Greeno (1988), the birds and worms problem has the same semantic structure as the traditional wording. Computer simulations (e.g., Kintsch & Greeno, 1985 and Riley & Greeno, 1988)
There are 5 birds and 3 worms. How many birds won't get a worm?

**Figure 36.**

"Won't Get" version of Compare 1

predict, in general, that problems of identical semantic structure require similar logico-mathematical knowledge. Therefore, if students can not solve the traditional wording they should not be able to solve the birds and worms rewording since they lack, for example, a part-whole schema which is claimed to be necessary to solve these types of problems. Hudson's (1983) results clearly mark the distinction between logico-mathematical knowledge and the role of problem wording which facilitates the connection between the text and that knowledge, as argued by Cummins (1991), Hudson (1983) and others:

"How many more than?" questions do not involve a lack of appropriate correspondence skills, but instead involve a misinterpretation of comparative constructions of the general form "How many ... [comparative term] ... than ...?" (Hudson, 1983, p89).

The difficulty with (the logico-mathematical) view is that it is not clear how a simple rewording can allow a child without sufficient part-whole knowledge to solve such problems. In other words, why should a simple rewording matter? One possibility is that the rewording simplifies the problem structure so that part-whole knowledge is not required to solve it, that is, the rewording provides key words or the like that immediately trigger a problem solving action. As Hudson (1983) was at pains to point out, however, this is not the case. Even 4-year-olds exhibited a sophisticated grasp of part-whole relations in their problem-solving procedures. The data instead seem to indicate that the knowledge is there, but is simply not accessed when problems are worded in certain ways (Cummins, 1991, p267).

Table 47 compares Hudson's (1983) solution probability results and similar results from DeCorte, Verschaffel et al. (1985) with the traditional wording of Compare 1. These results provide strong support for the use of a situational context that is familiar to a student as an aid toward grasping the quantitative relationships in the problem, as confirmed by Staub and Reusser (1991), Stern and Lehrndorfer (1992), and others.

The third potential rewording to make Compare 1 easier replaces the static relational language with action language which describes a hypothetical act of making the two sets equal, as shown in Figure 37.
Table 47.
Comparisons of ‘‘Won’t Get’’ versions and the traditional Compare 1

<table>
<thead>
<tr>
<th>Problem Wording (grade)</th>
<th>Hudson Data</th>
<th>DeCorte Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional (Kindergarten)</td>
<td>0.25</td>
<td>-</td>
</tr>
<tr>
<td>Won’t Get (Kindergarten)</td>
<td>0.96</td>
<td>-</td>
</tr>
<tr>
<td>Traditional (1st)</td>
<td>0.64</td>
<td>0.47</td>
</tr>
<tr>
<td>Won’t Get (1st)</td>
<td>1.00</td>
<td>0.70</td>
</tr>
<tr>
<td>Traditional (2nd)</td>
<td>-</td>
<td>0.76</td>
</tr>
<tr>
<td>Won’t Get (2nd)</td>
<td>-</td>
<td>0.90</td>
</tr>
</tbody>
</table>

This rewording is typically categorized as an Equalize problem and is often listed as a separate type of semantic structure (e.g., Carpenter & Moser, 1984). In this current categorization of problems, Equalize problems are listed within the Compare semantic structure, since Equalize problems are specific rewordings to facilitate the comparison of two disjoint sets. In the results of Carpenter, Hiebert and Moser (1981), first grade children who used blocks to solve the Equalize version shown in Figure 37 were successful 91% of the time as compared to the results of Riley and Greeno (1988) where children were suc-

Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans does Jacob need to find to have as many as Chrissy?

Figure 37.
‘‘Need to Find’’ version of Compare 1
cessful only 28% of the time.

In addition, the computer model predicts that the Equalize version is easier than the traditional wording. In the Equalize version, the phrase "How many does Jacob need to get" (along with the fact that Jacob already has a set) is an explicit reference to a Join situation. Also, the phrase "to have as many as Chrissy?" language is an explicit reference to the result of that Join. On the other hand, when the model solves this problem with the traditional wording, the model must infer both the Join action and the result of that action. In terms of predictions, the model predicts that 100% of first grade children can solve the Equalize version while the model predicts only 42% can solve the traditional wording.

11.2. Making Compare 1 Problems Harder

There currently is no hypothesis for making this problem more difficult. At present, this seems reasonable since this problem appears at a very high level of mathematical abstraction, i.e., the text is void of any situational language or language that implies the results of situations (e.g., left, altogether). A change of focus in the question, e.g., asking how many Jacob has less rather than how many Chrissy has more, is one possibility; however, the less than version is considered as a Compare 2 problem which is discussed in the next section.

In summary, real world situations and action language of the forms "won't get" and "need to get to have as many as" provide two potential versions which make Compare 1 problems easier. Table 48 summarizes a relative order of difficulty for these types of problems.

11.3. Making Compare 2 Problems Easier

Figure 38 shows Riley et al.'s (1988) Compare 2 problem. In Compare 2 problems, the cardinalities of two disjoint sets are compared. The question requests the difference between the sets with a focus

---

7 The traditional wording of this type of problem has typically used the phrase "less than" rather than "fewer than." Although "fewer than" is the correct use, the examples to follow use the traditional phrasing (i.e., less than).
Table 48.
Compare 1 reworded versions

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>“Won’t Get”</td>
<td>There are 5 birds and 3 worms. How many birds won’t get a worm?</td>
</tr>
<tr>
<td></td>
<td>“Need to Find As Many As”</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 [soda] cans. How many [soda] cans does Jacob need to find to have as many as [Chrissy/His cousin]?</td>
</tr>
<tr>
<td>hardest</td>
<td>“More Than”</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 [soda] cans. How many [soda] cans does [Chrissy/His cousin] have more than Jacob?</td>
</tr>
</tbody>
</table>

Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans does Jacob have less than Chrissy?

Figure 38.
Traditional Compare 2

on the smaller set, i.e., “how many less?” Compare 2 problems are in general as difficult as Compare 1. In the Riley et al. (1983,1988) results, solution probabilities were identical across grades for Compare 1 and 2, except for first grade where success on Compare 2 problems (17%) was lower than Compare 1 (33%).

Determining a set of “easier” wordings than the traditional wording of Compare 2 is not as straightforward as in Compare 1. There is no analog of Hudson’s (1983) “won’t get” version of Compare 1 for Compare 2. Hudson’s version is worded with a focus on the larger set and requests the number of items in the larger set which are “unmatched” or, for example, “won’t get a worm.” Rewording
Compare 2 with a focus on the smaller set (as opposed to the larger set) results in an Equalize "need to get" version as shown in Figure 39. Although this version emphasizes a Join situation which more closely resembles the traditional wording of Compare 1 rather than Compare 2, this "need to get" version is also included as one of the easier wordings for Compare 2.\footnote{It is an artificial and unnecessary distinction to claim that the Equalize Join wording is only helpful for children experiencing difficulty with the traditional wording of Compare 1. Clearly, the need to get wording in Figure 39 may also help students who are having difficulty with the traditional wording of Compare 2.}

It is also possible to replace the static relational language (how many less than) in the traditional wording of Compare 2 with action language which describes a hypothetical act of making two sets equal, as shown in Figure 40. This Equalize Separating-From version, unlike the traditional wording of Compare 2, focuses on the larger set (no pun intended), i.e., Sarah.

In short, there are two possible Equalize rewordings which can make the less than construction easier to understand: (i) keep the perspective fixed on the owner of the smaller set and present a hypothetical action which will add to that amount in order to make it equal to the larger amount or (ii) change the

---

Jacob has 3 soda cans. Chrissy has 5 soda cans. How many soda cans does Jacob need to get to have as many as Chrissy?

Figure 39.
Equalize Join version for Compare 2

---

Henry weighs 65 pounds. Sarah weighs 70 pounds. How many pounds does Sarah need to lose to weigh as much as Henry?

Figure 40.
Equalize Separating-From version for Compare 2

---
perspective to the owner of the larger set and present a hypothetical action which will decrease the amount of the larger set such that it is equal to the smaller set. As always, the use of related co-actors is included as a possible help.

11.4. Making Compare 2 Problems Harder

As discussed in Compare 1, there currently is no hypothesis for making these problems more difficult. Table 49 summarizes a relative order of difficulty for the less than difference-unknown types of problems.

11.5. Making Compare 3 and 4 Problems Easier

Compare 3 and 4 problems are presented in full in Chapter 5.

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>“Need to Find As Many As”</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 [(soda) cans]. How many [(soda) cans] does Jacob need to find to have as many as [Chrissy/His cousin]?</td>
</tr>
<tr>
<td></td>
<td>“Need to Lose As Many As”</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 [(soda) cans]. How many [(soda) cans] does [Chrissy/His cousin] need to lose to have as many as Jacob?</td>
</tr>
<tr>
<td>hardest</td>
<td>“Less Than” Traditional</td>
<td>Jacob has 3 soda cans. [Chrissy/His cousin] has 5 [(soda) cans]. How many [(soda) cans] does Jacob have less than [Chrissy/His cousin]?</td>
</tr>
</tbody>
</table>
11.6. Making Compare 5 and 6 Problems Easier

Figure 41 shows Riley et al.'s (1988) Compare 5 problem. In Compare 5 problems, a set with a known amount is related to a previously unmentioned set. The known amount (referred to as the compared set) is described as being "n more than" the unknown set (referred to as the referent set). The question requests the amount of the referent set.

Figure 42 shows Riley's (1988) Compare 6 problem. In Compare 6 problems, like Compare 5, the compared set with a known amount is related to an unmentioned referent set. In Compare 6, however, the compared set is described as being less than the referent set. The question requests the amount of the referent set.

The two alternate wordings (more than with unknown referent set, less than with unknown referent set) are generally referred to as inconsistent Compare problems, according to the terminology of Lewis and Mayer (1987). These two problems are inconsistent because the language used to describe the relation is inconsistent with the necessary arithmetic operation in each problem, for example, Compare 5 with

Chrissy has 5 soda cans. She has 2 soda cans more than Jacob. How many soda cans does Jacob have?

Figure 41.
Traditional Compare 5

Jacob has 3 soda cans. He has 2 soda cans less than Chrissy. How many soda cans does Chrissy have?

Figure 42.
Traditional Compare 6
more is "inconsistent" since that problem requires subtraction but more "implies" addition. Likewise, Compare 6 with less is "inconsistent" since that problem requires addition but less "implies" subtraction. As mentioned previously in Chapter 5, inconsistent language problems like Compare 5 and 6 are significantly more difficult than consistent language problems like Compare 3 and 4 (Lewis & Mayer, 1987; Nesher & Teubal, 1975; Riley & Greeno, 1988; Stern, 1993). In addition, recent eye-movement experiments show that even with students who can successfully solve both consistent and inconsistent problems, inconsistent problems (Compare 5 and 6) take longer to solve. In addition, these results show that a significantly greater amount of fixation time is spent on the relational sentences of inconsistent problems as opposed to the relational sentences of the Compare 3 and 4 consistent problems (Hegarty et al., 1992; Verschaffel et al., 1992).

While there is agreement that inconsistent problems are more difficult than consistent problems, there have been no experiments to date that specifically test for the relative order of difficulty within the two types of inconsistent problems, i.e., between the traditional wordings of Compare 5 and 6. The empirical data on children's solution probabilities show that these two problems are the most difficult of any of the other eighteen problems in Riley's (1988) classification and that Compare 6 (less than) is slightly more difficult on average than Compare 5 (more than). In eye movement experiments, Verschaffel et al. (1992) report that college students made more reversal errors (e.g., they added when the problem required subtraction) on Compare 6 type problems (41.6%) than on Compare 5 problems (21.6%). On the other hand, the computer model predicts that these two problems require identical loads on working memory and an equal number of inferences.

There exist two possibilities to make Compare 5 and 6 problems easier: the use of (i) Equalize language and (ii) related co-actors. With Equalize language, the static relational language (e.g., how many more less than) in the traditional wordings of Compare 5 and 6 can each be replaced with action language which describes a hypothetical act of making the two sets equal (e.g., how many does one need
to find/lose to have as many as the other). For all grade levels, the computer model predicts that the Equalize version for Compare 5 and 6 will improve solution probability, especially for the youngest grades.

11.7. Making Compare 5 and 6 Problems Harder

As discussed for the other Compare problems, there currently is no hypothesis for making these problems more difficult. Table 50 summarizes a relative order of difficulty for the more than and less than (reference quantity unknown) types of problems.

<table>
<thead>
<tr>
<th>Difficulty</th>
<th>Problem Type</th>
<th>Worded Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>easiest</td>
<td>&quot;If Find As Many As&quot;</td>
<td>Chrissy has 8 soda cans. If [Jacob/her cousin] finds 5 (more) [(soda) cans], [Jacob/her cousin] will have the same number [of (soda) cans] as Chrissy. How many [(soda) cans] does [Jacob/her cousin] have?</td>
</tr>
<tr>
<td></td>
<td>&quot;If Lose As Many As&quot;</td>
<td>Jacob has 3 soda cans. If [Chrissy/his cousin] loses 5 [(soda) cans], [Chrissy/his cousin] will have the same number [of (soda) cans] as Jacob. How many [(soda) cans] does [Chrissy/his cousin] have?</td>
</tr>
<tr>
<td>&quot;More Than&quot;</td>
<td>Traditional Compare 5</td>
<td>Chrissy has 8 soda cans. She has 5 [(soda) cans] more than [Jacob/her cousin]. How many [(soda) cans] does [Jacob/her cousin] have?</td>
</tr>
<tr>
<td>hardest</td>
<td>&quot;Less Than&quot;</td>
<td>Jacob has 3 soda cans. He has 5 [(soda) cans] less than [Chrissy/his cousin]. How many [(soda) cans] does [Chrissy/his cousin] have?</td>
</tr>
</tbody>
</table>

Table 50. Compare 5 and 6 (Reference Quantity Unknown) reworded versions
APPENDIX D

Implementation Details

The computer model software is written in Common Lisp. The EDUCE component totals approximately 23,000 lines of Lisp; the SELAH component comprises about 9,300 lines of Lisp, for a total of approximately 32,000 lines. The user has the option to run the model in interpretive fashion (i.e., load uncompiled code for development purposes) or first compile the Lisp. The size of the model when compiled is: EDUCE files (except external_lexicon): 262k; SELAH files: 152k.

Because the implementation has strictly followed the Common Lisp standard, the model executes on a number of platforms which support Common Lisp, including a RISC mainframe (Lucid Common Lisp/DECsystem, v4.0), SUN workstations (Sun Common Lisp, v4.0.1) and a Macintosh IIci (Macintosh Common Lisp; requires 4.5 MB disk space and 2 MB RAM).
BIBLIOGRAPHY

Cognitive modeling and intelligent tutoring. *Artificial Intelligence*, v42(1), 7-49.

*Inside Computer Understanding* (chapter 13), Schank, R.C. and Riesbeck, C.K. (Eds.),
Hillsdale, NJ: Lawrence Erlbaum.


A question of structure, in *Research Issues in the Learning and Teaching of Algebra*,
Council of Teachers of Mathematics, 57-59.

An integrated model of skill in solving elementary word problems, *Cognition and Instruction*, v1, 245-296.

*EDUCE: Explaining Discourse Understanding with Conceptual Expectations*. Unpublished Master’s Thesis. Department of Computer Science, University of New Hampshire, Durham, NH.


Semantic development: the state of the art, in *Language acquisition: The State of the Art*,


Effects of teaching sixth grade students to modify format variables of mathematics word problems. *Reading Research Quarterly*, v16(2), 175-199.


The role of understanding in solving word problems. *Cognitive Psychology*, v20, 405-438.

The role of rewording and context personalization in the solving of mathematical word problems. *Journal of Educational Psychology*, v83, 61-68.


The effect of semantic structure on first graders’ solution strategies of elementary addition and subtraction word problems. *Journal for Research in Mathematics Education*, v18, 363-381.


Influence of the semantic structure of word problems on second graders’ eye movements. *Journal of Educational Psychology*, v82(2), 359-365.


Fletcher, C.R. (1986).  
Understanding and solving arithmetic word problems: a computer simulation. *Behavior Research Methods, Instruments and Computers*, v17, 565-571.


Comprehension of arithmetic word problems: Evidence from students' eye fixations. *Journal of Educational Psychology*, v84(1), 76-84.


Hiebert, J. (1982).


Correspondences and numerical differences between disjoint sets. *Child Development*, v54, 84-90.


Learning from text. *Cognition and Instruction*, v3(2), 87-108.


Toward a theory of automatic information processing in reading. *Cognitive Psychology*, v6, 293-323.


Teachers’ knowledge of students’ knowledge in mathematics problem solving: Correlational and Case Analyses. *Journal of Educational Psychology*, v81(4), 558-569.


The computer’s effort after meaning: word problem solving as a process of text comprehension and mathematization. Research Report #4, Department of Instructional Psychology, University of Bern, Switzerland, December, 1987.


A computer-based strategy for personalizing verbal problems in teaching mathematics. 


Artificial intelligence and learning environments: preface *Artificial Intelligence*, v42(1), 1-6.


The role of presentational factors in understanding and solving mathematical word problems. Presented at the annual meeting of the American Educational Research Association, San Francisco; also Research Report #10, University of Bern, Department of Instructional Psychology, Switzerland.


Thompson, E. (1967).


Causal relatedness and importance of story events. *Journal of Memory and Language*, v24, 595-611.

The effects of causal relations and hierarchical position on the importance of story statements. *Journal of Memory and Language*, v27, 1-22.


Solving compare problems: An eye-movement test of Lewis and Mayer’s consistency hypothesis. *Journal of Educational Psychology*, v84(1), 85-94.
