This thesis presents our investigation of the problem of creating an intelligent tutor for solving algebra word problems that employs effective pedagogy. Among our major concerns are the problems of sufficient knowledge and proper knowledge organization. From student protocols and test papers, we derived sets of general purpose strategic rules and computation rules to formulate a cognitive model of problem solving. The model incorporates levels of strategic ability that form a progression from novice to expert. Additional analysis of the data was done to uncover, categorize, generalize, and explain the errors made by students when solving algebra word problems.

We implemented a computer program, al-Khorezmi, that exploits the model to trace a student's performance, catch errors and provide corrective feedback explaining the cause of the error, and give help that relates the help to knowledge that it has discussed with the student. The cognitive model emphasizes simulating correct, but not necessarily optimal, strategies that students employ. With our model, we can ensure that a student performs correctly when he employs his own approach before he is taught advanced strategies. Therefore, al-Khorezmi exploits an "understanding" of the
student's approach to problem solving. The model incorporates three levels of expertise in the form of strategic rule sets. Al-Khorezmi correlates the strategies with student actions to determine the student's problem-solving skill.

We studied the rules of our model to identify relationships important to learning and to interconnect the rules in an extended genetic graph. This knowledge structuring process led to observing that skill improvement can be accomplished (or explained to the student) by applying analogous rules that exploit more powerful methods. Furthermore, problem solving efficiency can be increased (or explained to the student) by combining rules into more powerful rules.

Rule simplification can be exploited to establish a simulation model of the generation of errors. Errors are classified as incorrect applications of correct rules, correct applications of incorrect or deviant rules, and incorrect applications of deviant rules. The latter category allows al-Khorezmi to recognize compound errors and account for some of the otherwise unexplained student errors found in our data.
Al-Khorezmi -
An Intelligent Algebra Tutoring System

by

Garland J. Bayley

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed December 16, 1988
Commencement June 1989
Date thesis is presented: December 16, 1988
© Copyright by Garland J. Bayley
December 16, 1988

All Rights Reserved
Acknowledgement

I would like to thank Bill Bregar for his advice, assistance, feedback, and patience as this project was carried out. Thanks also go to Tom Dietterich, Bruce D'Ambrosio, and Bill Burger for the insightful feedback that they gave me. Finally, thanks go to Art Farley who, with Bill Bregar, did the original development work on the cognitive model. Thank you all very much.

I would like to thank Mike Akerman and John Dranchak who supported my efforts by reducing my work load at the University of Portland and supported my need for computing resources.

I would like to thank Bob Licht and his students at Lutheran High School for volunteering to participate in the experimental use of al-Khorezmi.

I owe much to my wife, Grace, for her patience, and for the support she has given to me as I pursued this degree for 8 1/2 years.

Finally, I would like to acknowledge the contribution that Robert Day made by making mathematics come alive during my high school years. His contribution led eventually to this thesis.
# TABLE OF CONTENTS

Al-Khorezmi - An Intelligent Algebra Tutoring System 1

1. Introduction 1

1.1. Objectives 1

1.2. Background 3

1.3. Techniques 7
1.3.1. Pedagogy 8
1.3.2. Knowledge representation 8

1.4. Results 10
1.4.1. Modeling correct performance 12
1.4.2. Explanation 14
1.4.3. Modeling incorrect behavior 15
1.4.4. Implications for education 16
1.4.5. General results 17

1.5. Overview of al-Khorezmi 18

2. Pedagogy and its support 26

2.1. Defining what will be taught 26
2.1.1. General skill to be taught 26
2.1.2. Specific skills taught by al-Khorezmi 27

2.2. Pedagogical principles and their justification 29
2.2.1. The benefits of corrective feedback 29
2.2.2. The benefits of practice 30
2.2.3. Practice without corrective feedback 31

2.3. Techniques for supporting the pedagogy 31
2.3.1. Model development 32
   Advantages of the model 32
   Related work on student problem-solving models 33
   Model tracing 34
2.3.2. Extending the model's representation with genetic relationships 34
   Genetic relationships importance to a tutor 35
   Deviation relationships directly support pedagogy 35
   Genetic links and their relationship to learning 36
3. Problem-solving model

3.1. Model of student behavior
3.1.1. Strategic rules
- Weak forward-directed strategy
- Means-ends strategy
- Expert strategy
- Hill-climbing strategy
3.1.2. Computation rules
3.1.3. Problem difficulty
3.1.4. Characterizing student competence
- Distinguishing competency by observation of behavior
- Relation to the geometry tutor

3.2. Model enables meaningful response to student approach

3.3. Implementing a tutor based on the model
3.3.1. Simulating a student by employing the model only
3.3.2. Providing a preliminary interface for model tracing
3.3.3. Testing the interface with student protocols
3.3.4. Employing the model to provide hints
3.3.5. Verification of the model by experimental tutoring
- Demonstrating strategic model usage
- Demonstrating computation rule usage
- Demonstrating strategy determination
- Demonstrating compiled knowledge

4. The Genetic Graph

4.1. The importance of the genetic graph

4.2. Domain rule organization
4.2.1. Specialization/generalization
4.2.2. Analogies between specializations
4.2.3. Refinement/simplification
4.2.4. Analogies between refinements
4.2.5. Interdomain analogies
4.2.6. Derivation/origination

4.2.7. Combination
Pedagogical implications of combination rules

4.2.8. Deviation/correction

4.3. Strategy rule organization
4.3.1. Specialization/generalization
4.3.2. Analogies between specializations
4.3.3. Refinement/simplification
4.3.4. Analogies between refinements
4.3.5. Composition/component
4.3.6. Sophistication

4.4. Factors in the genetic graph demonstrating difficulty
4.4.1. Derived rules
4.4.2. Combined rules
4.4.3. Other difficulty factors

4.5. Providing a syllabus

4.6. Providing a student model

4.7. Providing tutorial capability
4.7.1. Providing a passive help system
   Use of passive help by students during online session
   Demonstration of help facilities not exploited by student subjects
4.7.2. Towards a more general tutorial

4.8. Relation to Wusor

5. Corrective Feedback

5.1. Introduction

5.2. Study of errors in recorded problem solving
5.2.1. Errors in simple wet-mixture proportion problems
5.2.2. Errors in forming and applying combined rules in wet-mixture problems
5.2.3. Mistaken analogies in wet-mixture problems
5.2.4. Errors in solving distance-rate-time problems
5.2.5. Comparison to other work

5.3. Uncovering mistakes through model tracing
5.3.1. Give-calculated-result option errors
5.3.2. Give-expression option errors
6. Implementation details

6.1. Strategic rule independence from domain computation rules

6.2. Student modeling

6.2.1. Modeling domain knowledge

6.2.2. Modeling strategic knowledge

6.2.3. Modeling misconceptions

6.2.4. Using the record of student knowledge and ability

6.3. Using the frontier in the student model for giving hints

7. Conclusion

7.1. Review of the goals

7.2. Evaluation of the results

7.3. Meeting Clancey's criteria*

7.4. Limitations of al-Khorezmi

7.5. Directions for further research

Bibliography

Appendices

Appendix A Computation Rules

Appendix B Experimental Usage
Appendix C
Effect of YAPS and its implementation

Appendix D Extending the interface

D.1. Carrying on an interaction 316
D.1.1. Creating discourse procedures for interaction 316
D.1.2. Formalizing discourse transitions 317
D.1.3. Discussing strategy with the student 318
D.1.4. Strategic and domain interaction based on learning relations 318

D.2. Pedagogical principles 319
D.2.1. Controlling combinatorial possibilities of induction 320
D.2.2. Employ genetic relationships for examples and explanation 321
D.2.3. Progress/regress appropriately for the student 322
D.2.4. Teach in the context of problem solving 323
D.2.5. Emphasize self-monitoring skills 323
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>D.3.</td>
<td>Design of a teaching system</td>
<td>324</td>
</tr>
<tr>
<td>D.3.1.</td>
<td>Lessons represented as disjuncts in the AND-OR graph</td>
<td>325</td>
</tr>
<tr>
<td>D.3.2.</td>
<td>Dynamic generation of the AND-OR graph</td>
<td>333</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Al-Khoremi's components with primary interconnections</td>
<td>19</td>
</tr>
<tr>
<td>1.2</td>
<td>Example run of al-Khorezmi</td>
<td>22</td>
</tr>
<tr>
<td>2.1</td>
<td>Example rules having genetic relationships (from Goldstein; 1979)</td>
<td>37</td>
</tr>
<tr>
<td>2.2</td>
<td>Problem solved by strategic rules applying CR1</td>
<td>46</td>
</tr>
<tr>
<td>2.3</td>
<td>Rule categories</td>
<td>47</td>
</tr>
<tr>
<td>2.4</td>
<td>Weak forward-directed rules (paraphrased in English)</td>
<td>50</td>
</tr>
<tr>
<td>2.5</td>
<td>Means-ends rules (paraphrased in English)</td>
<td>53</td>
</tr>
<tr>
<td>2.6</td>
<td>Expert rules (paraphrased in English)</td>
<td>54</td>
</tr>
<tr>
<td>2.7</td>
<td>Hill-climbing rules (paraphrased in English)</td>
<td>56</td>
</tr>
<tr>
<td>2.8</td>
<td>&quot;Structured&quot; English and YAPS forms of rule CR1</td>
<td>59</td>
</tr>
<tr>
<td>2.9</td>
<td>Problems from Groza (1978)</td>
<td>60</td>
</tr>
<tr>
<td>3.1</td>
<td>Protocol problems</td>
<td>44</td>
</tr>
<tr>
<td>3.2</td>
<td>Computer link student-tutor protocol</td>
<td>73</td>
</tr>
<tr>
<td>3.3</td>
<td>Simulation of student actions in Figure 3.10</td>
<td>75</td>
</tr>
<tr>
<td>3.4</td>
<td>Running a protocol through the interface</td>
<td>81</td>
</tr>
<tr>
<td>3.5</td>
<td>Giving hints based on the problem-solving model</td>
<td>86</td>
</tr>
<tr>
<td>3.6</td>
<td>Determining use of the expert strategy</td>
<td>91</td>
</tr>
<tr>
<td>3.7</td>
<td>Problem 1, Problem 2, and CR1 for specialization</td>
<td>95</td>
</tr>
<tr>
<td>3.8</td>
<td>Graph of refinements and analogies for wet-mixture</td>
<td>98</td>
</tr>
<tr>
<td>3.9</td>
<td>Structured English statements of CR1, CR8, and CR10</td>
<td>100</td>
</tr>
<tr>
<td>3.10</td>
<td>Structured English computation rules for demonstrating refinement analogies</td>
<td>102</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.5</td>
<td>Strategic rules FD2.1, FD2.2, and FD3.2</td>
<td>103</td>
</tr>
<tr>
<td>4.6</td>
<td>CR3 and CR8</td>
<td>105</td>
</tr>
<tr>
<td>4.7</td>
<td>Examples for strategic rule specialization</td>
<td>109</td>
</tr>
<tr>
<td>4.8</td>
<td>Graph of refinements and analogies for strategic rules</td>
<td>112</td>
</tr>
<tr>
<td>4.9</td>
<td>Procedure for solving word problems (Bello, 1985, p. 205)</td>
<td>114</td>
</tr>
<tr>
<td>4.10</td>
<td>Strategic rules ME1 and Goal Reporting</td>
<td>114</td>
</tr>
<tr>
<td>4.11</td>
<td>Strategic rules FD2.1 and FD2.2</td>
<td>116</td>
</tr>
<tr>
<td>4.12</td>
<td>Strategic rules ME3.1, ME3.2, and EX3</td>
<td>117</td>
</tr>
<tr>
<td>4.13</td>
<td>Simple mixture problems from intermediate algebra pretest</td>
<td>119</td>
</tr>
<tr>
<td>4.14</td>
<td>Mixture problems from section 1 midterm</td>
<td>122</td>
</tr>
<tr>
<td>4.15</td>
<td>Distance-rate-time problems from section 2 midterm</td>
<td>123</td>
</tr>
<tr>
<td>4.16</td>
<td>Specialization, derivation, and combination links for wet-mixture domain rules depicting levels of difficulty</td>
<td>127</td>
</tr>
<tr>
<td>4.17</td>
<td>Genetic links with explanatory text</td>
<td>131</td>
</tr>
<tr>
<td>4.18</td>
<td>Student DH's interaction with al-Khorezmi</td>
<td>135</td>
</tr>
<tr>
<td>4.19</td>
<td>Demonstrating traversal of links other than specialization</td>
<td>138</td>
</tr>
<tr>
<td>4.20</td>
<td>Strategic rules EX2.2, ME2.1, and ME2.2</td>
<td>143</td>
</tr>
<tr>
<td>5.1</td>
<td>Computation rules for wet-mixture problems</td>
<td>148</td>
</tr>
<tr>
<td>5.2</td>
<td>Two mixture problems with applicable computation rules</td>
<td>150</td>
</tr>
<tr>
<td>5.3</td>
<td>Strategic rules ME2.1, ME3.2, and ME4</td>
<td>157</td>
</tr>
<tr>
<td>5.4</td>
<td>Computation rules CR8, DCR1, DCR2, and DCR3</td>
<td>159</td>
</tr>
<tr>
<td>5.5</td>
<td>A portion of BM's problem-solving interaction</td>
<td>172</td>
</tr>
<tr>
<td>5.6</td>
<td>A portion of CW's problem-solving interaction</td>
<td>175</td>
</tr>
<tr>
<td>6.1</td>
<td>Flavors for wet-mixture problem schema</td>
<td>179</td>
</tr>
<tr>
<td>6.2</td>
<td>Flavors for distance-rate-time problem schema</td>
<td>180</td>
</tr>
</tbody>
</table>
Figure 6.3. Functions access and access-next-to-last........................................182
Figure 6.4. LISP expression which generates the description of CR3........183
Figure 6.5. Strategic rules FD2.1, FD3.1, and ME2.1........................................186
Figure 6.6. Handling misconceptions...............................................................189
Figure 6.7. Passive help for student DH (repeat of Figure 4.18)..................191
Figure 6.8. The active help system (excerpted from Figure 3.11).................194
Figure 6.9. Computation rules for wet-mixture problems (repeat of
Figure 5.1)........................................................................................................197
Figure 6.10. Strategic rules ME2.1 and ME2.2..................................................197
Figure 7.1. Al-Khorezmi's response to student mistake (from Figure
5.6)................................................................................................................206
Figure 7.2. Clancey's spectrum of student assessments...................................208
Figure C.1. Mix-in flavor db-utility and its methods.......................................313
Figure D.1. Part of the weak forward-directed strategy in an AND-OR
graph.............................................................................................................326
Figure D.2. AND-OR graph representing repetitive processing......................328
Figure D.3. Completed AND-OR graph of strategic rules...............................330
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Table 3.1</td>
<td>Responses to student approach</td>
<td>70</td>
</tr>
<tr>
<td>Table 4.1</td>
<td>Groza's tabular solution of a complex mixture problem</td>
<td>107</td>
</tr>
<tr>
<td>Table 4.2</td>
<td>Intermediate algebra pretest results</td>
<td>120</td>
</tr>
<tr>
<td>Table 4.3</td>
<td>Results from first midterm given to first section</td>
<td>122</td>
</tr>
<tr>
<td>Table 5.1</td>
<td>Categorization of errors in applying the correct rule</td>
<td>149</td>
</tr>
<tr>
<td>Table 5.2</td>
<td>Errors when employing the operator from a derived/original rule</td>
<td>151</td>
</tr>
<tr>
<td>Table 6.1</td>
<td>Modeled rule usage for user in Figure 3.11</td>
<td>196</td>
</tr>
</tbody>
</table>
In this thesis, we present a prototype intelligent tutoring system for algebra word problems. This system is named al-Khorezmi after the author of the first book on algebra, *Al-gabr wa'l-muqabalah*. Boyer (1968) quotes Robert of Chester's Latin Translation of this book to give the reason al-Khorezmi wrote it:

...to compose a short work on Calculating by (the rules of) Completion and Reduction, confining it to what is easiest and most useful in arithmetic, such as men constantly require in cases of inheritance, legacies, partitions, law-suits, and trade, and in all their dealings with one another, or where measuring of lands, the digging of canals, geometric computation, and other objects of various sorts and kinds are concerned (p. 152).

In short, al-Khorezmi's goal was to teach the applications of algebra, which is also one reason for including word problems in modern algebra textbooks.

1.1. Objectives

In this thesis, we investigate the problem of creating an effective, intelligent tutor for solving algebra word problems. In particular, we tutor the formulation of word problems into appropriate mathematical relationships. Because this is a rather open-ended problem, we focus on four subproblems that are crucial to solving this problem. These subproblems are:
1) to employ effective pedagogy,

2) to develop a cognitive model of this problem-solving task that is optimized for use in an intelligent tutor,

3) to develop a capability to meaningfully explain the relationship between new knowledge and the student's existing knowledge, and

4) to diagnose errors and provide corrective feedback by developing and employing a generative theory of bugs.

The major question addressed herein regards the inclusion and representation of knowledge necessary to adequately model problem solving by a student, and to successfully communicate that knowledge. As does Clancey, (1986a), we claim these problems to be more important than particular implementation techniques such as "rules" vs. "frames."

Pedagogy is an obvious problem, since we are investigating the development of a tutoring system. Without an effective means of interacting with the student, the tutorial knowledge of the system will be wasted.

The modeling problem is important, because the better the system understands the student, the better it can instruct the student. If the system has a model that matches student problem solving, then it can better handle misconceptions the student has at his own level before directing the student towards the desired level. If a student is more than one level of ability away from a desired level, a model that captures the intermediate level(s) can be employed to direct the student's progression through the intermediate level(s) to the desired level.

Relating new knowledge to the student's existing knowledge in a meaningful way is important, because such relationships facilitate explanation, and lead to understanding and retention. If new knowledge is presented without reference to existing knowledge, then the student is forced
to make the connections on his own. If he does not make the connections, then the knowledge may not be mastered or easily retained.

Lastly, being able to generate bugs that match the student's discrepant behavior is important, because it enables corrective feedback (which is good pedagogy) that can explain the origin of the bugs. Therefore, not only can the system state that the student made an error, it can state what the error was and, significantly, how the student formulated the error.

1.2. Background

Carbonell (1970) initiated the idea of applying artificial intelligence techniques to computer-assisted instruction when he built a geography tutor that exploited an information network to allow a mixed-initiative dialogue. Further preliminary work in intelligent computer-assisted instruction (ICAI) was done on the coaching of games, e.g., West (Burton and Brown, 1982) and Wumpus (Goldstein, 1979), with the objectives of modeling the student's knowledge, diagnosing misunderstandings, developing tutoring strategies, and demonstrating or explaining refined techniques. Games may provide a rich domain for investigating tutoring, but they are limited, since they do not fit well into the mainstream of the educational system's curricula. Other research has been carried out on building intelligent tutors for academic subjects (i.e. tutoring meteorology, electronic trouble shooting, and medical diagnosis, Brown et al., 1973; Brown et al., 1982; Clancey, 1979, respectively) which is necessary to further a goal of enhancing the educational system's curricula.¹ For this reason, this work is part of a project that has worked on algebra in general (Curtis, 1979; Lantz et al., 1983; Rapp, 1986).

¹For those interested in a more detailed introduction to artificial intelligence and tutoring systems, Wenger (1987) is a good introduction to the field.
Polya recognized the significance of algebra word problems when he wrote

the most important single task of mathematical instruction in the secondary schools is to teach the setting up of equations to solve word problems (1962, p. 59).

The intention here is the need to learn to employ symbols for quantities when working on real-world problems and to produce systems of equations that model the problem. Greeno (1978) recognized that skill development is done by developing skills at one level and then integrating them into higher level units.

Currently, when a student attempts to work on or review word problems on his own, he finds himself in a difficult situation. His textbook gives a cryptic statement of how to proceed and some examples with solutions. For example, Gobran (1984) gives the instructions for employing one variable:

Determine the unknown quantity and represent it with a variable. All other unknown quantities must be expressed in terms of the same variable. Then translate from the problem the statements relating to the variable into an algebraic equation. Solve the equation for the unknown and then find the other required quantities. Check your answer in the word problem, not in the equation (1984, p. 66).

and then follows with numerous examples. Provided he has the patience, the student will have to expend much time and effort figuring out how to apply each piece of information given in the textbook. Anderson (1983) bases his theories of learning on the assumption that students first encode new information declaratively, and then apply general interpretive procedures to this declarative knowledge to carry out the procedure necessary to accomplish the task at hand. This may explain why so much time and effort is required to learn to solve problems in this fashion. Without feedback, the student may develop weak and incorrect methods. It is in this type of situation that an intelligent tutoring system can provide valuable advice. It could guide and
tutor the student through the problem-solving process, point out the applicability of the necessary pieces of information, and allow him to practice the necessary skills, correcting him when he makes a mistake.

To address the problem of providing an optimal cognitive model, we chose to pursue two significant facets of cognitive modeling. We separate strategic knowledge from tactical knowledge, and we accommodate novice through expert behavior. Clancey (1983; 1986a) and Clancey and Letsinger (1984) determined that cognitive models employed in intelligent tutoring systems are more useful if they separate strategic knowledge from tactical knowledge. They found that this separation enables the program to instruct both strategy and tactics, which is an improvement over requiring the student to formulate his own strategies from problem-solving instruction that does not make a distinction between strategy and tactics.

Chi et al. (1981) and Larkin et al. (1980a) studied the difference between expert and novice processing on physics problems. They found that experts first engage in qualitative analysis of the problem to categorize the problem and limit search. After the qualitative analysis, the experts establish the formulas to be employed. Finally, they solve the equations. In contrast, novices first categorize by the problem entities—the things, either of known or unknown quantity, that are mentioned in the problem. After this categorization, the novices derive their formulas from the categorization in a means-ends fashion.

To address the problem of developing a capability to meaningfully explain the relationship between new knowledge and the student's existing knowledge, we investigated the applicability of Goldstein's (1979) genetic graph. The genetic graph represents the origins and development of knowledge by relating knowledge, which is encoded as rules, with
relationships that pertain to its genesis and development. These relationships enable the meaningful explanation of new knowledge in terms of its relationship to existing knowledge. The graph also aids modeling by providing a topology on which to locate frontiers of a student's knowledge, and the relationships can be employed to relate buggy rules to correct rules.

To address the problem of diagnosing errors and providing corrective feedback, we developed a generative theory of bugs. Brown and Burton (1978) note that student mistakes are not as random as one might think, but instead students often consistently follow the same "bugged" procedure again and again. We concur with their belief that:

A detailed model of a student's knowledge, including his misconceptions, is a prerequisite to successful remediation (p. 156).

One generative theory of bugs is given by Brown and VanLehn (1980) who put forward repair theory in which a student makes a repair to his procedure when he reaches an impasse. Sleeman (1984) proposes a related explanation, which he calls mis-generalization, where students infer mal-rules that fit the examples given them, but do not work in general. Matz (1982) explains algebra errors as extrapolations from base rules. These extrapolations employ a known rule inappropriately in a new situation or incorrectly adapt a known rule to a new problem.

Once an appropriate model of processing in a domain has been developed, it can be exploited to monitor student behavior. If the granularity of the knowledge in the model is appropriate, then the model can be employed by an intelligent tutoring system to trace the actions of a student while he is performing the task. Anderson et.al. (1985) have termed this technique model tracing. They employed it in their geometry tutor. Likewise, the LISP tutor of Reiser et al. (1985) is based on model tracing. However, matching the
performance of a model to problem solvers' actions in an intelligent tutoring system has been proposed before by Bregar and Farley (1980). Indeed, verifying the suitability of a model is done by matching human processing to processing by the model (e.g., Luger, 1981).

1.3. Techniques

In this section, we discuss the techniques that we employed to seek solutions to the problems that we are addressing. We considered the problem of pedagogy, which is somewhat independent of the other subproblems, and the problem of sufficient knowledge organized appropriately, which encompasses our interdependent subproblems of modeling, explanation, and bugs. It should be noted that we are not attempting to evaluate pedagogy in this thesis, but had to adopt some pedagogical basis to create any kind of reasonable tutorial system. However, the problems of sufficient knowledge and proper organization are pivotal here. Clancey (1983, 1986a) and Clancey and Letsinger (1984) have pointed out how knowledge organization and content has been a limiting factor for expert systems and intelligent tutoring systems. Therefore, we address the problems of organization and sufficiency with respect to representing
1) strategic and tactical knowledge, 2)
2) the progression from novice to expert,
3) explanatory relationships, and
4) erroneous behavior.

1.3.1. Pedagogy

We considered pedagogical issues that would be general in nature rather than domain or system specific. This led to two principles that are paramount to learning procedural skills. These principles are:

1) providing timely, corrective feedback improves performance more than simply noting errors, and
2) "practice makes perfect" (assuming that the correct skills are being practiced).

We hold these principles as basic to the initial development of al-Khorezmi. Other pedagogical principles have been accorded a secondary role and are incorporated into the design of an extended system for teaching an entire curriculum of solving algebra word problems.

1.3.2. Knowledge representation

To address the problem of sufficient knowledge organized appropriately, we studied the knowledge that students and teachers exploit to solve algebra word problems. We based our study on the following questions:

---

2Here we refer to our computation rules as tactical knowledge, because they describe procedures for solving problems. Clancey (1986a) refers to the strategic knowledge as the inference procedure and to the tactical knowledge as the general model.
3In Chapter 2 we justify the selection of these two principles.
4The design of the extended system and associated pedagogical principles appears in Appendix D.
5In Chapter 2 we discuss how solutions to the problems presented by these issues supports our pedagogical principles.
To what extent should strategic knowledge be separated from tactical knowledge (Clancey, 1983; Clancey, 1986; Clancey and Letsinger, 1984)?

How does novice performance differ from expert behavior (Chi et al., 1981; Larkin et al., 1980b; Johnson et al. 1981)?

How can we facilitate explanation by relating new knowledge to old knowledge in a meaningful way (Goldstein, 1979)?

What is the cause of student errors in performing procedural skills (Brown and Burton, 1978; Brown and VanLehn, 1980; Matz, 1982)?

For our study, we recorded protocols of students working on algebra word problems with the aid of a tutor. These protocols are of two types. Some are audio recordings of students working on problems with the assistance of a tutor by their side, and others are transcriptions into a file of a student trying, via a computer terminal link, to solve a problem sent by the tutor in a separate room. The second set of protocols were collected, because they are a better approximation to the form of communication that exists between a computer program and a user. We examined these protocols to identify the separation between strategic and tactical knowledge, to identify the progression of ability levels from novice to expert, and to aid in the development of a generative theory of bugs. From there, we employed this information to model a preliminary interface for al-Khorezmi.

For modeling problem-solving, we adopted the production rule formalism introduced by Newell and Simon (1972). Production systems have been widely employed to model intelligent behavior. From the protocols, we derived sets of general purpose strategic rules and sets of computation rules particular to different word problem domains. We distinguished levels of strategic ability that are based on the performance of the novice students and the expert tutors and associated a rule set with each level.6

---

6See Bregar et. al. (1986) for further discussion of the development of these rules and Chapter 3 for the most recent formulation.
Once we had established rule sets, we studied the rules to identify relationships between them that learning research (e.g. Michalski et al., 1986) has identified as being important to learning. Goldstein (1979) proposed the employment of several of these, calling them "genetic" relationships after Piaget's (1971) *Genetic Epistemology*, which is the study of the origin and development of knowledge. Finally, we employed both the protocols and test papers from an intermediate algebra class to uncover, categorize, generalize, and explain the errors made by students when solving algebra word problems.

We employed the results of our studies to implement a computer program, because it is through implementation in a program that artificial intelligence has traditionally tested its theories of intelligent processing (Newell, 1982). Al-Khorezmi takes the form of a system that exploits the model of student performance to trace a student's performance, catches errors and provides corrective feedback explaining the cause of the error, and gives help, when requested, by relating the help to knowledge that it has discussed with the student.

1.4. Results

We have constructed a cognitive model of student problem solving that separates strategy from tactics and further subdivides strategy into three levels of ability. The emphasis of this model building is simulating correct, but not necessarily optimal, strategies that students employ. This differs from other research projects that employ either a model of expert competence or a

---

7Chapter 4 presents these relationships and their usage in the tutorial system.
8Chapter 5 presents the taxonomy of errors uncovered and presents an explanation of their origin that fits the data.
model of the target behavior. With our models, we can verify that a student performs correctly when he employs his own approach before he is taught more desirable or expert strategies. Therefore, we claim that this model of student problem solving can yield an effective intelligent algebra word problem tutor. Employing this model, the tutor can monitor and respond to a student by exploiting an "understanding" of the student's approach to problem solving.

Because we have modeled levels of strategy that have overlapping components, we must discern the student's level of processing within these strategies. To do this, we have identified distinguishing characteristics of the strategies and programmed al-Khorezmi to watch for them.

Our model exploits as its basis for representation, a genetic graph that relates the components of our model. Employment of the genetic graph as a model led to the following observations:

- Improvement of skill in a formal domain such as mathematics does not come from refining simplified rules through several levels of improvement as Goldstein (1979) suggests it does for Wumpus. Because of the clear distinction between right and wrong in mathematics, the emphasis is on getting the one correct refinement instead of the partial refinements that are possible when learning to play Wumpus. These partial refinements to rules for selecting moves in Wumpus are not necessarily wrong, but they are not sufficient to yield the optimal move.

- A definition of analogy that we expanded beyond the one given by Goldstein (1979) can be employed to improve explanatory power, because it directly relates rules that otherwise are only indirectly related.

- The simplification relationship of the genetic graph can be exploited to establish a simulation model of the generation of errors. We constrain simplification to relaxing only the conditions for selecting one entity (an operand or an operator) in a rule so that another (incorrect) entity can be selected. This accounts for most of mistakes that we observed during attempts to apply a single computation rule. If we allow a rule to be simplified so that it selects one incorrect operator and one incorrect

---

9Wumpus (Yob, 1975) is an exploration game in which a player hunts a "Wumpus" in a cave that has interconnected rooms.
operand, then the remaining mistakes are accounted for. We consider these mistakes to be incorrect applications of a correct rule. A different way to constrain simplification is to allow the conditions that select all the operands for a rule to be relaxed by uniformly changing the type. This causes the new rule to incorrectly select all the entities. We consider the resulting rules to be incorrect or deviant rules that are applied correctly, and we relate them to their origin with a deviation link in the graph.

- The two methods of simplification can only be combined in one way—a deviant rule can be formed and then applied incorrectly. This third class of errors, deviant rules applied incorrectly, explains some of the otherwise unexplained student errors found in our data.

1.4.1. Modeling correct performance

We developed a rule-based model that separates strategic knowledge into three levels of competence and also separates strategic knowledge from tactical knowledge. Therefore, this model is a synthesis of previous developments in separating novice knowledge from expert knowledge and previous developments in separating strategic knowledge from tactical knowledge. Our strategic model incorporates a progression in ability from the novice level (weak forward-directed strategy) through the intermediate level (means-ends strategy (Newell and Simon, 1972)) to the expert level (expert strategy). The weak forward-directed strategy employs the known quantities of the problem to generate values for unknown quantities. The means-ends\textsuperscript{10} strategy creates formulas for the goal and, if necessary, identifies subgoals that must be found before the goal can be found. The expert\textsuperscript{11} strategy creates complex formulas that can be solved directly to find the goal. The tactical

\textsuperscript{10}We call this strategy means-ends, because it is the name employed by Larkin et al., (1980a) and others. Mathematics educators and artificial intelligence researchers may recognize this strategy as goal-directed problem solving or as backward chaining.

\textsuperscript{11}We also derive the use of the word expert for this strategy from Larkin et al. (1980a). Mathematics educators do not refer to the activity at this stage as problem solving. They refer to it as doing exercises, because it is exercising knowledge gained through experience with a particular mathematical domain. To mathematics educators, problem solving is solving previously unencountered problem types, and expert problem solvers are those people who are very good at solving such problems.
knowledge is encoded in computation rules. Computation rules relate the quantities of the problem in a computational relationship such as
\[
\text{distance} = \text{rate} \times \text{time}.
\]

Because we derived our model of problem solving from protocols of student work, we claim that it accurately models student performance. We obtained further substantiation of this claim by

- encoding the model in a program that is able to replicate student work found in our protocols and

- employing al-Khorezmi, which traces student problem solving by matching their problem-solving operations to the model, to experimentally tutor students who performed operations that the model describes.

The interface that al-Khorezmi employs does not provide the student with operations from any of the strategies nor with computation rules to apply unless he explicitly asks for help. Instead, al-Khorezmi provides general operations for the student to apply and determines the specific strategies and computations that he chooses. In the experimental use of al-Khorezmi with high school freshmen through seniors, all of the strategic operations (including those attempted on paper without the system) attempted by the students were modeled. Because of the range of students that employed it, al-Khorezmi traced students who worked at levels ranging from the novice level through the expert level. Furthermore, al-Khorezmi understood all of the correct computation rules that the students applied except one correct computation rule, which employed proportion setup techniques, that a student applied. Therefore, the experimental use of al-Khorezmi substantiated our computation rule model with the exception that one student employed a technique that we purposefully left out.

Deciding the level of competence of a student when tracing his problem-solving performance is complicated by an overlap in the
applicability of rules at different levels in the model. The thesis proposes and demonstrates in the implemented prototype that basing this decision on differences in strategic rule sets allows determination of which students have advanced beyond the lowest level of ability and to what level they have advanced.

In the progression from novice to expert, we model the strategic steps of the expert with rules that are compositions of rules from the preceding ability level, the mean-ends strategy. Therefore, the manner in which our model describes the advancement of a student from the means-ends level to the expert level of ability replicates Anderson's (1986) learning theory of knowledge compilation, which states that improvement in ability on procedural tasks occurs when the learner forms compositions of the rules that he applies to the task.

1.4.2. Explanation

The combination of the genetic graph with the problem-solving model illustrated that the genetic links do not link novice to expert behavior by successive refinements as Goldstein proposed they might. Therefore, we replaced refinement as the basis of learning with sophistication links in the graph to direct al-Khorezmi from rules to their counterparts at the next higher level of ability, with composition links to show when ability is improved through the formation of new strategic rules by composing rules that normally execute in sequence, and with combination links to show when ability is improved through a hierarchical combination of domain rules. The composition and combination links provide a basis for explanation by allowing new knowledge to be explained in terms of existing knowledge. To provide for explanation of the relationship between new knowledge and existing
knowledge that is represented by a sophistication link, we defined two new forms of analogy beyond those defined by Goldstein (1979).

Experimental use of al-Khorezmi demonstrated the utility of relating knowledge with genetic links when students exploited the help facility to present specializations (example applications) of rules to show them how to solve problems. Upon seeing these demonstrations of specializations, they made comments like, "I see now." To what extent these students gained insight into solving other problems needs to be shown by further experimentation.

1.4.3. **Modeling incorrect behavior**

Goldstein (1979) suggests two classes of errors--correct application of incorrect rules and incorrect application of correct rules. Of course there is another class of rule application--correct application of correct rules. This trichotomy strongly suggests one further class--incorrect application of incorrect rules. We argue that if a characterization of the generation of the two classes of errors suggested by Goldstein combines in only one way, and this combination generates previously unexplained errors in students' work, then we have some support for this theory. Although this generative theory of bug origin is not as well developed as the repair theory of Brown and VanLehn (1980), it has the advantage of generating most bugs whereas repair theory "only generates 21 of the observed 89 bugs." This is not to say that our theory gets an advantage from being so general that it can generate almost anything. Instead, we have constrained simplification to two forms, thereby restricting what the theory can generate. Further validation of our theory through the application of techniques such as *competitive argumentation* (VanLehn et al., 1984) is definitely desirable, but for the time being our theory
enables al-Khorezmi to respond to student errors with comments that are plausible explanations for the errors.

In the experimental use of al-Khorezmi, the representation of buggy knowledge proved quite successful. Seven of the twelve students that participated made one or more mistakes. The encoding of our generative theory of bugs, which we developed based on the two genetic relationships, simplification and analogy, enabled al-Khorezmi to explain all of the mistakes made. Complete coverage of the errors is particularly important, because corrective feedback is very important to our pedagogy.

Finally, studying bugs and bug correction processes illuminated a deficiency in our approach. We did not represent the self-monitoring (metacognitive) behavior that experts employ. It is highly desirable for students to learn this self-monitoring behavior. Although we did not represent this behavior so that it could be simulated and taught, we found ways to encode messages into our corrective feedback so that such behavior can be suggested to students when they make a mistake.

1.4.4. Implications for education

There are some implications for education that come about as a side effect of this work. The model's progression of ability levels is the basis for a theory of problem difficulty. This progression corresponds to the intuitive feeling of problem difficulty that mathematics educators have exploited in designing instructional materials.\(^{12}\) This correspondence supports the choices made in sequencing the progression of the strategic levels of the model.

\(^{12}\)The section on problem difficulty in Chapter 3 further develops this correspondence.
Demonstrating that the expert strategy is a compiled form of the means-ends strategy suggests that the emphasis of many textbooks on the expert strategy (e.g. Bello, 1985; Gobran, 1984; Groza, 1978; Phillips et al., 1983) be replaced by more emphasis on teaching the means-ends strategy. While both strategies can solve the same set of problems, the means-ends strategy works in general from a smaller knowledge base, whereas the expert strategy requires a large knowledge base of combined rules.

Our analysis of the errors that students made while setting up word problems gives a specific set of errors for educators to watch for in their students' work and gives a explanation for the generation of those errors. The generalization of this analysis into a generative theory of bugs holds even wider implications for the explanation of errors in procedurally oriented tasks depending on how well the theory applies to the learning of other tasks.

1.4.5. General results

The most general result is that a system incorporating the answers to our questions on modeling, explanation, and bug detection has been implemented as a prototype. Therefore, the feasibility of building such a system has been demonstrated. Complete answers to whether the proposals of this thesis and their implementation will make the most effective intelligent tutorial system can only be realized after much further investigation. Note that al-Khorezmi could be tested against more traditional script-based computer-assisted instructional systems for algebra word problems, but that will not answer questions concerning the proper organization of an intelligent tutorial system. At this point, the implemented prototype can only be employed to partially show or support some results. A major obstacle to showing more complete results is the absence of other systems that employ
different principles or different configurations of the same principles to serve as benchmarks.

1.5. Overview of al-Khorezmi

Figure 1.1 depicts the components of al-Khorezmi. We discuss these components below. The interface of al-Khorezmi accepts a student's problem-solving operations as input and employs a qualitative student model to trace these operations as he solves a problem. Clancey (1986a) defines a qualitative student model to be a description of a student's problem-solving procedure and its relationship to the concepts of the domain in which the problem solving is taking place. Anderson et al. (1985) have coined the term model tracing to describe tutors that employ such models to trace student work.
Figure 1.1. Al-Khoremi's components with primary interconnections.
As the interface performs its model tracing it assesses of the student’s work. It employs the problem representation to assess the correctness of statements he makes about the problems, and it employs the qualitative student model to assess the correctness of operations he carries out. When he makes mistakes in statements about the problem or in applying computation rules, al-Khorezmi provides corrective feedback. The corrective feedback provides not only a statement of the error in applying a computation rule and a statement of the correct application of the rule, but also how the error was generated, which the bug generation rules of the generative theory of bugs uncover. Al-Khorezmi maintains a quantitative student model by recording the number of times the student applies each strategic and computation rule.

The student can ask for help on how to use al-Khorezmi and receive explanations of the operations al-Khorezmi allows him to carry out. The student can exploit the passive help system to take the initiative to explore the database of descriptions of the rules that are in the qualitative student model. The student does his exploration by following the genetic links that connect rules having epistemological relationships. The student can employ the active help system to let al-Khorezmi take the initiative and present hints that it thinks he can follow through on, because either the quantitative student model records that he has previously performed the suggested operation, or the operation is linked by a genetic link to an operation he has previously performed. Because al-Khorezmi can apply the qualitative student model to solve the problem at hand, it can determine an appropriate next step to suggest as a hint to a student.

In Figure 1.2 below, we give excerpts from a run of al-Khorezmi on a simple problem. The characters in **boldface** are typed by the user and the portions in *italics* are comments added to explain portions that have been
omitted or to discuss the processing of al-Khorezmi. These designations of users' interactions and added comments will be employed throughout the remaining chapters when output from al-Khorezmi is included.
To begin a session al-Khorezmi states a problem and presents a menu which lists the quantities in the problem. For example, in the problem below menu entry a) is for the quantity the amount of the 20% solution, menu entry b) is the amount of alcohol in the 20% alcohol solution (note the word alcohol on the left of the line), and menu entry f) is the proportion of water in the 20% alcohol solution. Given this menu the student can refer to each quantity by its letter and specify values for it or include it in formulas. Although percent is mentioned here, al-Khorezmi employs only proportions to save processing time. Percents are easy to incorporate later.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol are in the vat?

\[
\begin{array}{llll}
\text{alcohol} & \text{b) amount} & \text{c) proportion} & \text{d) percent} \\
\text{water} & \text{e) amount} & \text{f) proportion} & \text{g) percent} \\
\end{array}
\]

v) full proportion constant 1.0  
w) full percent constant 100.0  
y) other value

The menu of problem quantities is followed by a menu of operations that the student is allowed to do. For example, option 2) allows the student to inform al-Khorezmi that he has found the value of one of the quantities in the problem statement and option 5) allows the student to enter an algebraic expression relating the problem quantities.

1) note goal  
2) note constant  
3) specify a variable  
4) give calculated result  
5) give expression  
6) solve  
7) help-hints  
8) help-menu-options  
9) help-algebra  
10) exit-problem

Option? 8 This is the option employed to get explanations of menu entries.

Which option would you like to have explained? 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10? 2

The problem statement mentions some values that are known and therefore constant for the problem. This option should be used to tell the system which constants you have noted. You will be prompted for Quantity? to which you should answer with the letter from the menu of quantities. You will be prompted for Value? to which you should answer with a number given (or implied) in the problem.

Preceding each request that the user select an option the menu that appeared first is redisplayed in updated form. Any values that become known are displayed in the menu.

Option? 2 Quantity? a Value? 10
OPERATION VERIFIED TO BE CORRECT

---

Figure 1.2. Example run of al-Khorezmi.13

---

13The contents of the figure were generated by the system and displayed on a 132 column computer terminal. However, the layouts of this figure and others have been edited to a more readable layout for presentation in this thesis.
At this point the menu was redisplayed in its entirety with the following line changed as shown.

   a)  amount-20%-solution 10

Option? 2 Quantity? c Value? 0.2
OPERATION VERIFIED TO BE CORRECT

The menu is redisplayed with the .2 in place. From now on you can assume that the menu was just displayed whenever you see - Option?

Option? 8

Which option would you like to have explained? 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10 ? 9

This option allows you the explore the strategies and relationships used in solving the current problem.

Option? 9

This is the passive help system that lets the user explore the knowledge base and its application.

Would you like help on:

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu

? 2

A rule applicable to your request is cr10: WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF THE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SUBSTANCES

Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

? 7

A rule applicable to your request is cr3: THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION

Figure 1.2 (continued)
Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

amount-alcohol-20%-sol( ) = proportion-alcohol-20%-sol( ) * amount-20%-solution( )

Would you like to see values with that equation (y or n)? y

amount-alcohol-20%-sol(Unknown) = proportion-alcohol-20%-sol(0.2) * amount-20%-solution(10)

Would you like to see another example (y or n)? n

At this point the user can perform the .2*10 or have al-Khorezmi do it for him with options 5 and 6 and he is finished with the problem.

Figure 1.2 (continued)

We implemented al-Khorezmi in YAPS, (Allen, 1983) a production system monitor, and Flavors in Franz Lisp (Allen et al., 1983) running under Unix on a VAX 11/750. It contains slightly more than 300 YAPS rules and employs Flavors for the schemas that store objects and relationships associated with problem types. Currently only wet mixture (Mayer, 1981) problems are handled in the tutor. Additionally, schemas and computation rules for motion and current (Mayer, 1981) distance-rate-time problems have been developed. These have been employed by the problem solver separately to solve distance-rate-time problems at all three ability levels. The tutor has been kept independent of the problem type so that if the rules for distance-rate-time problems were encoded into their various uses in the tutor, the tutor would be able to solve distance-rate-time problems too. It is the generality of the strategic rules that allows computation rules and associated entity descriptors
for additional problem types to be developed and added to the knowledge base enabling additional problem types to be solved and tutored.

We employed rule-based programming supplemented by the structured objects available in YAPS (Allen, 1983) for the entire implementation of a prototype, but do not claim this is the best way to proceed. Indeed, we claim that lessons learned in the implementation of the prototype can lead to better implementations. The importance of doing an implementation is to demonstrate the efficacy of our answers and therefore the correctness of our hypothesized solutions.
2. Pedagogy and its support

2.1. Defining what will be taught

We begin by defining clearly what we are attempting to tutor. This will provide a background for the following sections which discuss our pedagogy, and the components of our work that support it. At the end we state the things that we are not attempting to tutor in this pass on the problem.

2.1.1. General skill to be taught

Our interest is tutoring problem solving for solving algebra word problems. When students solve algebra word problems, they typically start from a written statement of a problem. This statement gives constant values for some attributes of some of the entities involved and asks for the unknown value of some attribute of an entity. The problem entity attributes will typically have some well known computational relationship between them such as

\[ \text{distance} = \text{rate} \times \text{time}. \]

After analyzing the problem statement and entities included, students typically produce a sequence of operations that lead to the answer, produce formulas that can be solved to realize the answer, or produce a combination of these. Their actions will depend on their level of sophistication. They may range in sophistication from having difficulty searching out and establishing each relationship they employ to directly applying specialized formulas applicable only to the problem type at hand.

We wish to tutor the steps necessary to proceed from the expression of the problem and its constants to determining the requested unknown, which
we refer to as the goal. These steps include converting relationships implied in the problem to explicit formulation as operations or formulas and carrying out the operations or solving the formulas. Performing these steps involves employing general strategies and specific domain relationships. Our goal is to assist students to progress from novice to expert by tutoring more powerful general strategies. For this purpose we have developed models of strategic competence that represent different ability levels. If our goal is accomplished, a student with these strategic abilities will be able to enter a physics curriculum, for example, with the strategic skills necessary to perform adequately as a learner of physics. Chi, et al. (1981), and Larkin, et al. (1980a) have shown that capable learners of physics employ the means-ends strategy on problems. Means-ends is our intermediate ability level.

2.1.2. Specific skills taught by al-Khorezmi

We want to enable students to recognize the goal(s) of the problem, to correctly instantiate formulas expressing domain relationships, to recognize the need for subgoals when necessary, to employ symbols and formulas as a solution aid, to recognize when formulas can be combined as a solution aid, and to recognize when the problem is solved. These goals reflect facets of problem-solving that we have modeled with three sets of strategic rules that delineate the three ability levels.

For example, in the problem

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

The student needs to recognize the goal to be a value for the amount of pure alcohol to be added. From this he could instantiate a formula

\[ x = y - 10 \]
for

\[
\text{amount of the 10\% solution} = \text{amount of the 70\% solution} - \text{amount of pure alcohol}
\]

In this formula he should recognize that finding \( y \) (the amount of the 70\% solution) will help him find \( x \) (the amount of the 10\% solution) and therefore \( y \) can be thought of as a subgoal. Recognition of the ability to combine the relationship

\[
\text{amount of a substance in a solution} = \text{proportion of the substance times amount of the solution}
\]

with

\[
\text{amount of a substance in one solution} + \text{amount of the substance in another solution} = \text{amount of the substance in the composite of the two solutions}
\]

to produce the formula

\[
.2(10) + 1x = .7y
\]

for the problem at hand aids in solving the problem. Finally, at this point it should be recognized that the problem is essentially solved.

We endeavor to take a problem solver entering at the lowest level of ability through the intermediate level and have him achieve the highest level of proficiency. While tutoring students on strategic techniques, particular domains of application will have to be employed. Therefore, by necessity we must tutor employment of computational relationships from such domains. The side effect will be that students will receive practice at working with relationships such as

\[
\text{distance} = \text{rate} \times \text{time}
\]

that are applicable in the "real" world.
2.2. Pedagogical principles and their justification

We were mainly concerned with showing students how the use of appropriate strategies can help them solve problems more effectively. We investigated two major pedagogical principles that assist effective tutoring:

- Correcting a mistake with an explanation of the error improves performance over simply noting that the error has occurred.
- Practice of correct skills is necessary to ingrain them.

We incorporated these principles in building al-Khorezmi. The corrective feedback is given by tracing the steps the student takes and correcting errors as soon as they occur. The practice is possible by loading in new problems from prepared exercise sets. Rapp (1986) has created a system that will parse problems and create our representation so that a student could enter problems he is interested in. However, we have not tried to combine the two systems. We consider the best approach to be one where the system makes knowledgeable decisions on the next problem to present. Our design of a complete teaching system includes the pedagogical principles relating to the selection of problems and topics to teach.14

2.2.1. The benefits of corrective feedback

Tait et.al. (1973) show the benefits of corrective feedback in computer-assisted instruction. Guthrie (1971) shows feedback only provides its benefit on incorrect answers. Therefore, we conclude that feedback will have its greatest effect on the least advanced pupils. We claim that the capability of an intelligent computer-assisted instruction system to provide corrective feedback explaining why the answer is wrong will lead to even better student

---

14The design of the complete teaching system is given in Appendix D.
performance. This is in contrast to a computer-assisted instruction system that only notes that an answer is wrong. For these reasons we place a major emphasis in al-Khorezmi on the discovery and explanation of errors.\textsuperscript{15}

Bloom (1984) shows that private human tutors provide the most effective instruction. We attribute much of the effectiveness of human tutors to their ability to give corrective feedback. This is supported by Bloom's report that private tutors benefit the poorer students the most. Thus, one of the primary justifications for building an intelligent tutoring system is the corrective feedback it can give on an individual basis. We claim that the reason the poorer students receive the most benefit is because they make the most mistakes and the tutor is able to explain their mistakes, or at least explain the proper techniques in response to their mistakes.

2.2.2. The benefits of practice

According to the learning theories of Anderson (1986), expertise is gained through knowledge compilation by the processes of \textit{proceduralization} and \textit{composition}. Proceduralization is the process whereby a repeated usage of a general rule with specific data causes a new rule to form to handle situations with that data automatically without having to match up the general rule to the data. Composition of rules happens when a learner applies two or more rules in sequence often enough that he forms a new rule embodying the composition of the rules. These processes take place during the performance (practice) of a skill. Anderson's work shows that skills are not obtained directly from instruction but from performance of the skill. The verification of the processes of proceduralization and composition as a correct model of

\textsuperscript{15}The handling of errors will be discussed in Chapter 5.
skill refinement explains the accepted belief that improvement comes with practice. Keith (1982) reports that the amount of time spent on homework (practice) results in improved performance which lends support to the adage "practice makes perfect."

2.2.3. **Practice without corrective feedback**

Brown and Burton (1978) and Chaiklin (1984) show that typical student errors in the performance of a procedural skill are not random. Instead, they have buggy procedures that they follow consistently. This further motvates the development of a "practice field" where students can practice a skill and be corrected when they err. This would help in preventing the development of buggy procedures.

The possibility of providing a "practice field" with a knowledgeable coach is a highly attractive one. Currently, only the most talented students (the ones with the least need) end up with a coach in activities such as "Odyssey of the Mind," which is a problem-solving competition in which schools participate.

2.3. **Techniques for supporting the pedagogy**

We support our pedagogy with a knowledge base consisting of a production rule model based on student-tutor protocols and linked together with genetic relationships. The reasons for supporting our pedagogy with a model of student behavior and the genetic graph are given in the following sections.
2.3.1. **Model development**

We studied student-tutor protocols to develop a model of problem-solving skill with different ability levels. Our model has three levels--novice, intermediate, and expert. The novice level models the naive behavior of a beginning student. The intermediate level models goal-directed problem solving. And, the expert level models techniques applied by experienced problem solvers that are compilations of the intermediate level rules. Student-tutor protocols helped formulate the model, because the tutors followed the students at their level of performance and tutored more sophisticated techniques which were employed in the expert model. Furthermore, the students provided the typical performance that can be expected at low levels of ability. In effect the students' performance provided a *descriptive* model whereas the tutors' performance provided a *prescriptive* model (cf. Anderson and Skwarecki, 1986).

**Advantages of the model**

A model that can simulate student behavior from novice to expert provides several advantages. The system, by employing the model, can interact with the student at his level of competence just as our protocols have shown tutors doing. Such a system has a representation of the learner's knowledge on which it can base its record of a learner's knowledge instead of basing it on an expert's knowledge. Student modeling based on an accurate model of student behavior can help a system record what knowledge the student has by observing its use and can help the system infer some possibilities for what the student does not know by noting the knowledge it has not observed the student employing.
A model based on student performance should model the errors observed in student performance. This model can explain why the student action was in error by relating it to the correct action that should have been taken and it can explain how the student may have generated such erroneous performance originally. It is conceivable that the misconceptions novices hold will differ from the misconceptions a student at an intermediate level will hold. If this is true, a model that represents different levels of proficiency will be able to represent these misconceptions. How well the model satisfies these modeling problems is a major factor in determining the sufficiency of the model and how much of an advantage it gives.

Finally, the model limits the corpus of information that can be tutored. If the model only represents expert knowledge, then it will only be able to tutor knowledge at that level, which may not be appropriate for the student being tutored. Whereas, if the model represents several levels of problem-solving ability, then a model-tracing system that employs the model will be able to tutor at the student's level and to tutor improvements in ability in small steps, which may be more appropriate for the student when learning complex procedures.

**Related work on student problem-solving models**

Early intelligent tutoring systems such as SOPHIE (Brown et al., 1974) were based on an expert problem solver. Goldstein (1979) expressed the fundamental limitation of such an approach: "A learner's knowledge is not necessarily a subset of an expert's knowledge." Since that time, other work in intelligent computer-assisted instruction has been based on detailed modeling of student performance (e.g., Anderson et al., 1985; Reiser et al., 1985).
Model tracing

The extensive modeling of the student's capabilities enables building a system that exploits model tracing (Anderson et al., 1985). A model-tracing tutor matches the steps a student takes to the steps the model specifies. Because the tutor follows each step a student takes, it can provide corrective feedback at the time he makes a mistake, and it can obtain knowledge of his ability by recording the steps he takes. Therefore, it is particularly powerful in helping to attain our primary pedagogical principles. Additionally, having a record of the knowledge and capabilities of the student supports the advantages gained from having more knowledge of the student's ability.

2.3.2. Extending the model's representation with genetic relationships

In the previous section we outlined the importance of an in-depth model of student-expert performance. We listed several advantages of this technique. We gain further advantages by incorporating genetic relationships between the rules that form the knowledge structures. Genetic relationships were defined by Goldstein (1979) to represent the evolution of procedural knowledge. We employ the genetic relationships defined by Goldstein and add to them other relationships that are being studied in learning research because of the importance of these new relationships to the evolution of knowledge. In the rest of this section we explain why the genetic relationships can improve the tutor.
Genetic relationships importance to a tutor

The model tracing paradigm allows an accurate record to be kept of a student's knowledge, but it does not tell what topic to take up next with the student. To determine the next topic, the system can follow the genetic links to a new topic on the frontier of the student's knowledge. The graph represents the evolution of knowledge. Thus the system can be more or less confident of whether a student has mastered various topics based on whether the topic is well within the boundaries, on the frontier, or beyond the frontier of the student's knowledge. The relationships also provide connections that allow new or corrective information to be presented or explained in terms of known information.

Deviation relationships directly support pedagogy

Deviation links connect incorrect rules to their correct counterparts. Note that incorrect use of correct rules and incorrect use of incorrect rules are handled by procedural attachment, not by deviation links. Connecting incorrect rules to their correct counterparts allows al-Khorezmi to not only say that something is wrong but to say why. If the system has some experience working with the student and it has recorded observation(s) of the student applying the correct counterpart to the incorrect rule, then the system can call the student's attention to how he misapplied a known rule. Currently, al-Khorezmi does not make this distinction (although it does have a record of the students work). Instead, al-Khorezmi proceeds by stating the correct rule along with an explanation of why the rule the student applied is incorrect and how he may have constructed that rule. The deviation links directly support the corrective feedback principle of our pedagogy.
Therefore, they are very important to the development of a system embodying a principle of corrective feedback.

**Genetic links and their relationship to learning**

In the following sections we define the genetic links of Goldstein (1979) and the links that we added. We then relate them to their use in mathematics education to demonstrate their utility. In addition, we describe how the processes these links specify have been studied in research on learning. We make no attempt to verify their value, but we expect that if the processes are important to learning, they will be valuable in tutoring.

**Goldstein's genetic links**

The generalization/specialization link is defined as

R' is a *specialization* of R if R' is obtained from R by instantiating one or more of the variables of R. *Generalization* is the inverse relation.\(^1\)

Figure 2.1 gives example rules from Goldstein (1979). In these example rules, R2.2 specializes to R2.2B and R2.2P by instantiating the general term (variable) warning to squeak and draft respectively, and instantiating the general dangerous cave set D+ to the specific cave sets, bat dangerous cave set B+ and pit dangerous cave set P+, respectively. When all variables of a rule are instantiated with constants, then the relationship is between a general rule and a concrete example. Since extensive use of rules given with examples exists in mathematical instruction, this relationship has obvious importance. The generalization/specialization relationship corresponds to the *turning*

---

\(^{16}\)Goldstein (1979) gives the following definition:

R' is a *generalization* of R if R' is obtained from R by quantifying over some constant [sic]. *Specialization* is the inverse relation (p. 56).
constraints into variables selective generalization rule that has been defined by the study in artificial intelligence of learning through generalization from examples (Michalski, 1983). Anderson's (1986) proceduralization produces, from an old rule, a new rule that is related to the old rule by the specialization relationship.

R1.1: Add N to fringe cave set F.
R2.2: If warning, add N to dangerous cave set D+.
R2.2B: If squeak, add N to bat dangerous cave set B+.
R2.2P: If draft, add N to pit dangerous cave set P+.

Figure 2.1. Example rules having genetic relationships (from Goldstein; 1979).

Simplification/refinement links are defined as

R' is a refinement of R if R' manipulates a subset of the data manipulated by R on the basis of some specialized properties. Simplification is the inverse relation (Goldstein; 1979, p. 57).

In Figure 2.1, R2.2 is a refinement of R1.1, because it adds the condition: if warning. This condition causes R2.2 to manipulate a subset of the data manipulated by R1.1. In the production rule formalism, the refinement process refers to adding conditions, and possibly the associated actions, to a rule. Mathematics has examples of the refinement process because it has general rules that either do not always work or are not always the optimum choice. Therefore, the applicability of these rules must be refined. The simplification/refinement relationship corresponds to the dropping condition selective generalization rule that has been defined in artificial intelligence learning research (Dietterich and Michalski, 1983). Mitchell et al. (1983)
exhibit the refinement of optimum choice. They demonstrate improvement in performance through refinement of the applicability of mathematical operators such as the chain rule. Anderson (1986) demonstrates the employment of refinement (which he calls discrimination) to correct the applicability of rules. In Chapter 5 we will demonstrate the importance of proper refinement of the applicability of rules in explaining how errors are generated.

Analogy links are defined as

R' is analogous to R if there exists a mapping from the constants of R' to the constants of R (Goldstein; 1979, p. 56).

Here we interpret constants as: those values inserted by the instantiation process that is employed to form specializations of general rules. Therefore, two different specializations of the same rule are analogous. In Figure 2.1, R2.2B is analogous to R2.2P, because there exists mappings from squeak to draft and from bat dangerous cave set B+ to pit dangerous cave set P+. Analogies are employed tacitly all the time in mathematics instruction due to the extensive use of examples (instantiated rules).

**Supplemental genetic links**

We have not taken the narrow view of analogies that Goldstein has. Instead we extended the definition of analogies to include production rules which have a mapping between conditions. This allows linking together rules that are each linked by refinements to a third rule. In the study of analogy in learning (e.g., Part Four, Michalski et al., 1986) an even wider view of what is an analogy is often taken. Furthermore, structural analogy is considered by Anderson et al., (1984) to be of primary importance in learning procedural skills.
Combination links are a one-to-many link type we added to the genetic links of Goldstein. We define combination links as

\[ R' \text{ is a combination of } R \text{ and } R_1,R_2,\ldots,R_n \text{ if the action parts of } R_1,R_2,\ldots,R_n \text{ produce components of the action part of } R; \text{ and } R' \text{ is formed by replacing the components of the action part of } R \text{ by the actions of } R_1,R_2,\ldots,R_n \text{ that produce them, and by replacing the conditions of } R \text{ by the conditions of } R_1,R_2,\ldots,R_n \text{ that select the components of } R \text{ produced by } R_1,R_2,\ldots,R_n. \]

For example, when an airplane flies with the wind, the standard computation rule \( (R) \)

\[ \text{distance} = \text{net-rate-of-travel} \times \text{time} \]

can be combined with the rule \( (R_1) \)

\[ \text{net-rate-of-travel} = \text{wind-speed} + \text{airplane-speed} \]

to obtain the combination \( (R') \)

\[ \text{distance} = (\text{wind-speed} + \text{airplane-speed}) \times \text{time}. \]

In this example, the conditions which select net-rate-of-travel are to be replaced with the conditions that select wind-speed and airplane-speed and the net-rate-of-travel in the action part of the rule is replaced by the sum of wind-speed and airplane-speed. Replacing the wind-speed or the airplane-speed with another expression makes the hierarchical nature of the combination process more apparent. As a further example of the hierarchical nature of the combination process, consider the analogous composition of functions without recursion. For example, given

\[ z = f(x,y), \]
\[ x = g(u,v), \text{ and} \]
\[ u = h(s,t), \]

then by composition

\[ z = f(g(h(s,t),v),y), \]

which has \( g \) subordinate to \( f \) and \( h \) subordinate to \( g \).
Anderson's (1986) theory of knowledge compilation in learning employs an analogous process called composition (Lewis, 1978) to explain improvements in performance. Neves and Anderson (1981) provide a definition of composition that we re-express as

R' is a composition of R₁ and R₂ if C₁ and C₂ are sets of conditions of R₁ and R₂, respectively; A₁ and A₂ are sets of actions of R₁ and R₂, respectively; the condition of R' is C₁ \times₁ (C₂ - A₁); and the action of R' is A₁ ∪ A₂; where \times₁ is the relational algebra natural join applied to the result of applying the two sets of conditions to the database; and - A₁ removes any condition made true by applying A₁. Repetition of this process is applied to form compositions of more than one rule.17

Notice that composition results from a combination of rules executed in succession, whereas a combination link represents a hierarchical combination of expressions.

Wasson (1985) defines a genetic graph link type, component link, that serves as the inverse relationship to combination and composition. The component link is defined as

R' is a component of R if R' is a necessary element of R (p. 48).

The final link that we have defined is the derivation/origination link which relates rules that have an inverse relationship between them. However, we intend more than just an inverse relationship because we designate one of the rules as the base and the other rules are derived from it. We define the derivation/origination link as

R' is a derivation from R if the output of R' is an input to R, the output of R is an input to R', and the calculation of R' is derived from the calculation of R by solving for the new output. Origination returns a rule to its base form.

17This definition is quite different from the formal one that Neves and Anderson (1981) say the process "looks more like," because their formulation fails when three or more rules are composed. This definition was derived from their description of the process of composition.
For example, we designate the base rule of distance-rate-time problems to be:
\[ d = r \times t, \]
where \( d \) is the output and \( r \) and \( t \) are the inputs. Letting \( r \) be the new output, we solve to obtain the derived rule:
\[ r = \frac{d}{t}. \]
The derivation/origination relationship is easily seen to be important to mathematics, which has inverses as an important concept.

2.4. Related skills not incorporated

We have not dealt with tutoring general word-problem solving. There are problems for which our techniques are not appropriate. In the following sections we further define what we have done by enumerating things we have not done.

2.4.1. Solution of equations

Although we are tutoring the formulation of equations from word problems, we are not tutoring the process of solving those equations. See Lantz et al. (1983) for a system that tutors the process of solving equations. The strategic rules of our model indicate when to manipulate equations to solve or eliminate a non-goal unknown. Therefore, al-Khorezmi performs actual solving actions automatically when requested by the student. If the student does not request equation solution at the proper time, then the system can suggest it.
2.4.2. **Hill-climbing**

Our model includes a *hill-climbing* or systematic *guess-and-check* strategy, which is not incorporated in the tutor. This strategy models the problem solver who reaches an answer by testing successively better approximations.

2.4.3. **Other guess-and-check methods**

The hill-climbing strategy mentioned above falls under the category of guess-and-check methods (Butts, 1985). Other guess-and-check methods are systematic consideration of all possibilities and inductive examination of all cases. These techniques are often applicable to problems that are not solvable by employing equations with symbols for unknowns. Recently there has been much emphasis on problem-solving and the different approaches that are applicable to different types of problems. For example, see Silver (1985).

The setup of equations with unknowns represented as symbols is a general strategy that works for a broad base of problems. This is a skill required of many academic and nonacademic endeavors and therefore warrants our study. Other problem-solving strategies such as guess-and-check methods also warrant study, but are not addressed here.

2.4.4. **Reading comprehension**

The specialized vocabulary and its translation is another aspect of algebra word problems that causes difficulties for students. Stiff (1986) and others have worked on this problem in the standard classroom setting. In this thesis we are not trying to develop any specific techniques for this problem, but the related work by Rapp (1986) could serve as a point of departure for
such development. However, we do expect that students will improve their reading comprehension through practice in a tutoring environment.

2.4.5. Pictorial representation

Gould and Ginzer (1982) developed a system that exploits computer graphics to animate "time-distance-rate" problems. They tested whether employing such animation will improve the teaching of problem solving for "time-distance-rate" problems. They found that students' ability to draw problem representations improved, but they could not show improved problem-solving performance. They came to the conclusion that improving students' understanding requires a system that knows the subject matter and models the students' knowledge.

2.4.6. Setting up proportions

In developing an intelligent tutoring system, we worked primarily with wet-mixture problems (problems in which questions are asked concerning liquid solutions that contain specified percents of particular substances). One technique that is often taught for these problems is the use of proportions, an equality between ratios. Instead of incorporating this rather specialized strategy, we choose to concentrate on the general strategic rules, because they can be employed in other word problem domains.
3. Problem-solving model

3.1. Model of student behavior

This chapter presents our model of problem solving. This model was developed by examining verbal protocols of students and tutors working together on algebra word problems (Figure 3.1 gives some of the problems for which we collected protocols). The model presented here is a slight revision of the model described in Bregar et al. (1986). In particular, the model emphasizes the nature of qualitative processing (Larkin, 1977)—problem-solving behavior concerned with the establishment and subsequent solving of a symbolic representation of a problem. General strategic knowledge is separated from domain-specific computational knowledge; both are implemented as rule-based systems that allow declarative access, while providing control over the level of detail of the represented knowledge.

1. Tom drives at the rate of 40 mph for 3 hours. How far does Tom drive?
2. Tom has just mixed 3 gallons of alcohol with water to produce a 20% alcohol solution. How much solution does he have?
3. Time drove 250 miles on one day and 300 miles on the second day. He drove for 6 hours the first day, but only five the second day. What was his average speed while driving?
4. Tom has 12 gallons of a 25% alcohol solution. How many gallons of a 40% alcohol solution must he add to obtain a solution which is 30% alcohol?

Figure 3.1. Protocol problems.

\[18\] Portions of this chapter are excerpts from Bregar et al. (1986) and Bayley et al. (1987).
The model operates on a structured, symbolic representation that is the result of reading a problem statement, classifying the problem into one of a set of known types such as the overtake, opposite direction, and round trip distance-rate-time of Mayer (1981), instantiating predefined schemata associated with that problem type, and incorporating relevant information from the problem statement. Rapp (1986) reports on a system that reads problems in different domains and creates this representation. From the point at which problem entities have been defined and instances of schemata created to represent them, al-Khorezmi executes in a rule-based environment where rules match facets of the schemata. Two sets of rules interact to solve problems. The strategic rules articulate strategies observed in our student-tutor protocols, whereas the computation rules reflect typical expressions of formulas for a problem domain. This separation of knowledge provides generality of behavior as well as opportunity to tutor on domain-specific or general problem-solving topics independently.

The strategic rules fall into four categories with the two predominant categories being the rule-selection category and the solving-action category. By examining the current problem-solving state, rules in the rule-selection category select a computation rule to apply. In their base form, computation rules compute an output attribute from a pair of input attributes. For example, because the input attributes of CR1 have known values in the problem in Figure 3.2, strategic rule FD2.1 selects computation rule CR1. FD3.1 (from the solving-action category) then calculates the computation attribute of CR1 to produce the output attribute--22 gallons of total solution.
If you mix 12 gallons of a 10% alcohol solution with 10 gallons of pure alcohol, how many gallons will be in the total solution?

**FD2.1:** Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

**FD3.1:** If all input attributes of a bound computation rule have known values then perform the computation rule.

**CR1:** The amount of a solution is equal to the sum of the amounts of its parts.

- **Structure:** Composite solution C from solutions S1 and S2
- **Constraints:** S1 not equal S2
- **Input values:** amount(S1), amount(S2)
- **Output value:** amount(C)
- **Computation:** amount(S1) + amount(S2)

Figure 3.2. Problem solved by strategic rules applying CR1

With our model of student problem-solving behavior, we model the learning process by tracing problem solving and observing the use of new rules. This is only possible with a complete model of correct behavior ranging from novice to expert. Our model has three sets of strategic rules (the weak forward-directed, the means-ends, and the expert) that form a progression from novice to expert. Therefore, in addition to acquiring new knowledge by learning the computational relationships of additional domains, we see the student gaining expertise by progressing through the sets of strategic rules. We have ordered the progression of strategy rules sets by their power in solving problems. Because it employs model-tracing, al-Khorezmi reacts to and records the level of student problem solving without demanding that students completely master one strategy before moving on to the next.
3.1.1. Strategic rules

The strategic rules are independent of a particular domain and are therefore the primary concepts to be tutored. The primary purpose of a strategic rule determines in which of four categories it belongs. Figure 3.3 names the four categories and shows the flow of execution between them. In the following sections we present the weak forward-directed, means-ends, expert, and hill-climbing strategies that we discovered in student-tutor protocols. Within each strategy we number the rules according to their category numbers given in Figure 3.3.

Figure 3.3. Rule categories.

We assume that a learner begins by knowing how to solve word arithmetic problems, which involves applying some of the rules in the weak
forward-directed strategy. Kintsch and Greeno (1985) present a processing model that incorporates both the text-comprehension and problem-solving aspects of word arithmetic problems. Derry and Hawkes (1987) are developing an intelligent tutoring system based on this model. Because the algebraic use of symbols for unknowns is incorporated in our model, there are additional strategic rules that incorporate symbol use in the weak forward-directed strategy.

After learning to solve problems in the weak forward-directed manner, the student advances through the means-ends and expert strategies. While the student increases his knowledge and strategic ability, he retains old knowledge and abilities. Therefore, because ability acquired at an earlier stage is retained, not all rules exhibited by a problem solver at a particular level are repeated in the strategic rules for that level.

**Weak forward-directed strategy**

The weak forward-directed strategy solves problems of limited difficulty with algebraic techniques on which limits are placed. It is weak because there is little or no goal-directed behavior. It is forward-directed because known values are employed to compute new values. And, it solves problems of limited difficulty because it allows only equations with one unknown.

Figure 3.4 gives the five rules in the weak forward-directed strategy. The reader is encouraged to look over these rules before proceeding and to do likewise when each new rule set is introduced. The five rules in the weak forward-directed strategy come from three of the four categories of rules related to strategies. The first rule belongs to the goal-reporting category, the second and third rules belong to the domain rule-selection category, and the fourth and fifth rules belong to the solving-action category. These categories
are numbered beginning with 1, 2, and 3 respectively. The remaining category, *goal-establishment*, which first appears in the means-ends rules will be numbered beginning with 4.

One might note that two strategies could be distinguished here as rules FD1, FD2.1, and FD3.1 are sufficient to solve problems not requiring symbolic mathematics. However, since we are working with algebra students and they have been taught the use of variables, we do not make distinction between what would be considered pre-algebraic and algebraic capabilities in our strategies.

There are two rules in the domain rule-selection category, FD2.1 and FD2.2. FD2.1 selects a domain rule that computes an unknown as its output. FD2.2 chooses a domain rule having an unknown as input. This second rule is necessary to account for students who have not yet established a representation for the algebraic reformulations of a rule by legal algebraic manipulations, or who prefer to work with the base rule form. There are also two rules in the solving-action category. FD3.1 carries out the computation bound by FD2.1. FD3.2 applies algebraic manipulation to solve for the unknown in the equations bound by FD2.2.
FD1: If the problem goal attribute has a value then note the solution and report the result.

FD2.1: Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

FD2.2: Given a set of attributes with known values then select a rule with those attributes as an input and an output and determine consistent entity bindings with one input unknown

FD3.1: If all input attributes of a bound computation rule have known values then perform the computation rule.

FD3.2: If a bound computation rule has one unknown which is an input then solve for the unknown (isolate it on the left-hand side).

Figure 3.4. Weak forward-directed rules (paraphrased in English).

Means-ends strategy

Means-ends (Newell and Simon; 1972) is a search strategy that compares the current state to the goal state. It then computes the difference between these states, and selects for further exploration those operators that make the largest reduction in the difference between these states. Our means-ends strategy, like that of Larkin et al. (1980b), chooses equations that contain the goal variable (versus those that do not) to be the operators that make the largest reduction in the difference between the goal state (the goal variable being known) and the current state. These equations may have more than one unknown. Therefore, the means-ends strategy goes beyond the weak forward-directed strategy by adding goal-directedness and the establishment of subgoals from relationships that have more than one unknown. When applied in this manner, means-ends generally works backwards by starting from the goal and finding expressions to compute the goal. When these expressions for
computing the goal have additional unknown(s), then subgoals are established for the unknown(s) and processing is continued again in a means-ends fashion.

In addition to a change to goal-directed processing, another change in problem-solving behavior also occurs. This is a phenomenon which we have termed **spontaneous computation**. Spontaneous computation refers to observed behavior where students mention the values of quantities not given as constants in the problem. They do this without giving any indication of how they derived the values. We have not explicitly implemented this behavior, but we simulate it by firing FD2.1 and FD3.1 in succession. The students apparently have formed a rule which is the composition (Lewis, 1978) of rules FD2.1 and FD3.1. This is the first of several examples of the use of composition that we observed in improved processing by problem solvers and supports Anderson's (1986) contention that composition is a primary component of knowledge compilation.

Because a student reaches the means-ends strategy level only after accomplishing processing at the weak forward-directed strategy level, he is also capable of working in a forward direction. Therefore, he will often mix forward processing with backward processing. This is similar to the problem solving exhibited in geometry theorem proving (Anderson et al., 1985). The relationship between these two projects will be examined in section 3.1.4.

The means-ends strategy has rules (given in Figure 3.5) from each of the same categories as the weak forward-directed strategy and also has a rule from the category for goal-establishment. The first rule, ME1, belongs to the goal-reporting category and is employed to propagate when the value of a subgoal attribute has been found and needs to be propagated upwards in the goal tree.
The two rules, ME2.1 and ME2.2, in the domain rule-selection category are like the related rules in the weak forward-directed strategy. ME2.2 is also necessary here to account for students who have not yet established a representation for variants of a rule, or who prefer to work with the base rule form.

There are also two rules in the solving-action category, ME3.1 and ME3.2. ME3.1 takes a bound computation rule with a single unknown as one of its inputs and solves for it by isolating it on the left-hand side. Notice how this enables rules selected by ME2.2 to be solved. ME3.2 takes two bound computation rules and eliminates a common non-goal unknown.

The final rule, ME4, is a goal-establishment rule. It takes a bound computation rule with unknown(s) in addition to the goal unknown and makes those unknown(s) subgoals.
ME1: If a problem subgoal attribute has a value then note the solution and report the result and propagate the result upward in the goal tree.

ME2.1: Given a goal attribute then select a rule computing its value as output and determine consistent bindings.

ME2.2: Given a goal attribute then select a rule having it as input and determine consistent bindings.

ME3.1: Given a goal attribute and an equation (a bound computation rule) with the goal variable then solve the equation for that variable (isolate it on the left).

ME3.2: Given two equations having a nonprimary goal variable in common then combine equations eliminating the nonprimary goal variable.

ME4: Given a bound computation rule with unknown nongoal attributes and an unknown goal attribute then make the nongoal attributes be goal attributes.

Figure 3.5. Means-ends rules (paraphrased in English).

**Expert strategy**

The expert strategy is characterized by the immediate selection of domain rules that can be applied to solve the problem. These domain rules are often more specialized than those applied by the other strategies. For example, the problem

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

can be solved by

\[ .2(10) + 1x = .7(x+10) \]

---

19Recall from chapter 1 that people employing this strategy have sufficient experience with the problem type to formulate a solution technique immediately.
It can be seen from the example, that such domain rules are specialized to particular situations. Figure 3.6 gives the expert strategy rules which consist of two domain rule-selection rules and a solving-action rule. The first selection rule, EX2.1, selects domain rules such as the one in the example with the problem goal as the only unknown. The second selection rule, EX2.2, selects a domain rule based on a previously selected domain rule and its nongoal unknown. For example, in the problem above, once the formula

\[10 + x = y\]

is bound, EX2.2 would select

\[.2(10) + 1x = .7y.\]

Notice that EX2.2 is a composition of means-ends rules that set up subgoals and then select rules based on them. In the previous example, ME4 would make \(y\) a subgoal, and ME2.2 could bind the second formula given above. The final rule, EX3, solves two equations in two unknowns. This is a composition of ME3.2 which combines equations to eliminate a non-primary goal unknown and ME3.1 which solves for a goal unknown.

---

EX2.1: Given an unknown goal attribute then select a combined computation rule employing only it and determine consistent entity bindings.

EX2.2: Given an unknown goal attribute and a bound computation rule employing it and another unknown then select another computation rule with those unknowns and determine consistent entity bindings.

EX3: Given two equations with the same two unknowns then solve for the two unknowns.

Figure 3.6. Expert rules (paraphrased in English).
Hill-climbing strategy

The hill-climbing strategy (or systematic guess-and-check) can be characterized as a heuristic search over the value space of the goal. The key aspect of this strategy is the interaction between rules HC3.1 and HC3.2 (see Figure 3.7). HC3.1 will never be invoked unless HC3.2 has been, previously. Rule HC3.2 looks for a computation rule such that it has an unknown goal variable on the right-hand side, and all other attributes have known values. Rather than solving for the goal variable algebraically, the rule stipulates that a guess be made for the goal attribute's value and the computation of the rule is performed. If the guess is correct, HC1 fires and the goal is known. Otherwise, rule HC3.1 will continue to fire until an answer is found (or time runs out). The adjustment of value produces the hill-climbing effect, increasing or decreasing the value as appropriate to move the value closer to the goal value.

Close examination of this strategy will show that it only works when an expression is bound, employing domain rule-selection rules, with only one unknown. It works on a problem such as

What is the amount of solution when there are 3 gallons of alcohol in a 20% alcohol solution?

The student may guess 10 gallons and check 20% of 10 against the 3 finding it to be too low. The student would then proceed to make better guesses checking them by taking 20% of them and checking against 3. However, more difficult problems require that two unknowns exist in the expression with the goal variable. Therefore, when faced with the problem

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?
The student may guess 10 gallons, which gives 20 gallons for the 70% solution.
Now, verification must be done with a second equation in two unknowns or with a sequence of computations that reflect what that second equation would relate. The verification in this example could be done with

\[ .2(10) + 1(10) = .7(20). \]

By being goal-directed, hill-climbing relates to the means-ends strategy. It would apply rule ME2.2 to select rules with unknown goal attributes to which to apply its solving-action rules HC3.1 and HC3.2. Finally, upon determining the correct answer the hill-climbing strategy would report the answer with goal-reporting rule HC1.

---

**HC1:** If the problem goal attribute has a correct value then note the solution and report the result.

**HC3.1:** Given a goal attribute with an incorrect value and the incorrectness was determined by a bound computation rule then adjust the value of the goal variable in the appropriate direction and perform the computation rule and check the correctness of the goal attribute value.

**HC3.2:** Given a bound computation rule with unknown goal attribute input and all other attribute values known then assign the goal attribute a reasonable value and perform the computation rule and check if the assignment is correct (matches the known output).

---

**Figure 3.7.** Hill-climbing rules (paraphrased in English).

---

**3.1.2. Computation rules**

A computation rule represents a valid computational relationship among the attributes of entities in a domain. It does this by relating in a formula the attributes of the entity schemata and relational schemata. Entity
schemata are structured representations of objects found in a given class of problems, and relational schemata indicate how problem entities are interrelated according to semantically meaningful relations of the domain. For example, the basic entities of wet-mixture problems are substances. The entity schemata for substances have the form

```
substance
name: <string>
proportion-value: <decimal>
amount-value: <decimal>
type-value: <substance-type>.
```

The relational schemata specify the correspondence between objects in a problem statement to entity schemata and state the key relationship between them. In distance-rate-time problems, one such relation is the "opposite-direction-relation" between moving object entities. The relational schemata for opposite-direction-relations have the form

```
opposite-direction-relation
first-object: <entity-pointer>
second-object: <entity-pointer>
distance-apart: <decimal>.
```

The formula can be viewed in two ways: as an operation that can be carried out immediately when the inputs are all known and as an equation that can be involved with other equations in equation solving procedures. Constraints in the computation rules on the attributes in the schemata provide important matching information to the problem-solving system as it attempts to bind entity attributes to computation rule variables. Each rule has five parts: structure, constraints, inputs, outputs, and computation or formula (Figure 3.8 gives an example of one of the rules from the wet-mixture problem domain in
"structured" English along with its counterpart from the YAPS program). The structure part describes a subset of the relational schemata associations; the structure and constraints are, in effect, the conditional aspects of the rule. Constraints specify restrictions on bindings of attributes to variables named in the structure part. Inputs are attributes of the entities whose values are involved in the computation; the resultant value is that of the output entity.

The actual computation given in the computation part of a rule can involve arbitrary constants, as well. Sometimes the rule is better viewed as having a formula and no output. This occurs in rules that are typical of experts who generally proceed by solving equations rather than by performing calculations. The example in the earlier section on the expert strategy demonstrates how experts employ the computation rule CR1 (given in figure 3.8) to write down the formula

\[ 10 + x = y \]

with the computation part of the computation rule written on the left-hand side of the equals, and with the output part of the computation rule written on the right-hand side of the equals. Mayer (1982) finds that for simple problems, writing out formulas instead of directly performing the computation interferes with and slows down problem solving. For difficult problems, however, formulas or equations are necessary. This finding supports our decision to view the actions of these rules in two ways. A statement of the computation rules for the domain of wet-mixture problems is given in Appendix A.
"Structured" English statement of CR1

CR1: The amount of a solution is equal to the sum of the amounts of its parts.

Structure: Composite solution C from solutions S1 and S2
Constraints: S1 not equal S2
Input values: amount(S1), amount(S2)
Output value: amount(C)
Computation: amount(S1) + amount(S2)

YAPS rule for CR1

(defun cr1
  (process binding)
  (bind-rule-name cr1)
  (goal . -)
  (entity solution-mix -solution1)
  (entity solution-mix -solution2)
  (entity solution-mix -solution3)
  (entity mixture-relation -relation)
  test
  (= -solution1 (« -relation 'first-solution)) ;These tests implement the structure
  (= -solution2 (« -relation 'second-solution)) ;part of the rule and assure the
  (= -solution3 (« -relation 'composite-solution)) ;constraints are true.
  -->
  (let ((-inst (make-instance
                'computation-rule-instantiation
                'rule-name 'cr1
                'left-side (make-instance
                'expression
                'operator 'plus
                'operands (list (make-instance
                                 'quantity-description
                                 'entity -solution3
                                 'field 'amount-value)))
                'right-side (make-instance
                'expression
                'operator 'plus
                'operands (list
                                 (make-instance
                                 'quantity-description
                                 'entity -solution1
                                 'field 'amount-value)
                                 (make-instance
                                 'quantity-description
                                 'entity -solution2
                                 'field 'amount-value))))
       (<- -inst 'process-unknowns 'no-subgoal))))
  (installp 'cr1)

Figure 3.8. "Structured" English and YAPS forms of rule CR1.
3.1.3. Problem difficulty

Algebra word problems vary in difficulty. The textbook by Groza (1978) has three sections dealing with different facets of mixture problems, each reflecting either a different facet of the problems or an increase in difficulty in the problems. Figure 3.9 gives sample wet-mixture problems from sections 6.5, 6.6, and 6.7.

Section 6.5
What is the volume of acid in 40 cc. of a 5% acid solution?

Section 6.6
What is the volume of mixture obtained when 20 cc. of water is added to 40 cc. of a certain acid solution?
What is the volume of water added to 50 cc. of a certain acid solution to obtain a mixture whose volume is 75 cc?

Section 6.7
How many quarts of pure methanol should be added to an antifreeze containing 16% methanol to obtain 18 qt. of an antifreeze containing 30% methanol?
How much cream containing 35% butterfat should be added to milk containing 4% butterfat to produce 155 qt. of half-and-half containing 12% butterfat?

Figure 3.9. Problems from Groza (1978).

The weak forward-directed strategy can be employed directly to solve any of the first three problems. The fourth problem could be solved by the weak forward-directed strategy but with some difficulty. The key would be to notice that pure methanol has no water. Therefore, the amount of water would be zero. This shows the amount of water in the 16% solution to be the same as the amount of water in the 30% antifreeze solution. Therefore, problem solving proceeds as follows:

\[ 70\% = 100\% - 30\% \]
\[ 12.6 \text{ quarts water} = 0.7(18 \text{ quarts of the 30\% solution}) \]
12.6 quarts water in the 16% solution = 12.6 quarts water in the 30% solution - 0 quarts water in the pure methanol

84% = 100% - 16%

15 quarts of the 16% solution =

12.6 quarts water in the 16% solution .84

3 quarts of pure methanol = 18 quarts of the 30% solution - 15 quarts of the 16% solution

The means-ends strategy can solve this problem in an analogous way, by starting from the goal and reasoning backwards. The backwards reasoning proceeds as follows:

The amount of pure methanol can be obtained if the amount of 16% solution is known. This can be obtained if the amount and proportion of water in the solution is known. The proportion of water can be obtained from the percent of methanol, and amount of water can be obtained if the amount of water in the pure methanol and the amount of water in the 30% solution are known. There is no water in pure methanol, and the amount of water in the 30% solution can be obtained from the percent of water in the 30% solution and the amount of solution. The percent of water in the 30% solution can be obtained from the 30%.

From this goal setting, the computation would proceed as above. Notice that we are not claiming that students actually employ either of these solution paths. Instead our protocols show students employing a combination of the two.

Means-ends also allows the problem to be solved in a straightforward manner not involving the water. Because it is possible to employ more than one variable and to create subgoals, the necessary formulas can be instantiated solely in terms of quantities of methanol. In this case the following formulas could be created:

\[ x + y = 18 \]

\[ z = .16 \times x \]
\[ w = .3(18) \]
\[ z + y = w \]

where \( x \) is amount of the antifreeze containing 16% methanol, \( y \) is the amount of the pure methanol, \( z \) is the amount of methanol in the 16% antifreeze solution, and \( w \) is the amount of methanol in the 30% antifreeze solution. The same technique can be applied to solve the fifth problem, but it is not optimal.

The expert strategy could solve the fourth problem by setting up the equation

\[ .16(18 - x) + x = .3(18) \]

or by first setting up the equation
\[ x + y = 18 \]

and then finding another equation in the same two unknowns which in this case is
\[ .16 x + 1 y = .3(18) \]

3.1.4. Characterizing student competence

The weak forward-directed strategy characterizes problem solvers who make little use of symbols for unknowns. They are generally limited to employing one symbol at a time to represent unknowns. The students at this level can usually solve the first three problems given in the previous section. Few, if any, would find the key to solving the fourth problem with this strategy.

The means-ends strategy characterizes problem solvers who have learned to make better use of symbols in algebra, but who have not practiced with the use of symbols enough to become experts. Because they have not combined strategy and computation rules to make the new rules that allow experts to proceed directly to a solution, these problem solvers work backwards
from the goal searching for relationships that will satisfy their goals. The
students at this level can solve the first three problems. The fourth and fifth
problems provide a greater challenge, but with forward generation of obvious
quantities and working backwards from the goal, students can find a meeting
of the forward and backward processing allowing solution of the problem.
This distinction between the means-ends strategy and the expert strategy is
similar to that reported in Larkin et al. (1980a) for novices and experts
working on physics problems.

The division of the problem-solving model into three levels of ability
forms a theory of strategic competence in algebra word problem solution. If
the level of ability of a student can be determined, then attempts to predict the
success or failure of that student on different problems can be made.

**Distinguishing competency by observation of behavior**

The problem-solving model, and therefore the theory, does not define a
clear-cut distinction stating exactly when a student is exhibiting a specific
level of competence. This is because a student's observable problem-solving
behavior is insufficient for distinguishing which strategy rule he is applying.
The results a student provides, the formulas he writes, and the equation
solving operations he performs, make up the portion of his behavior that is
observable. Any other characteristics that are important to determining
which strategy rule a student applies, such as his goals, the direction of his
reasoning, and whether or not he is making combinations of rules, must be
inferred, unless he makes a special effort to make these characteristics
explicit.

When the student makes no indication whether he is working forward
from the problem constants or backwards from the problem goal, it is
impossible to determine his strategy in simple problems. For example, the problem

If you mix 12 gallons of a 10% alcohol solution with 10 gallons of pure alcohol, how many gallons will be in the total solution?

can be solved by considering that the amount of alcohol in the total solution is the goal, and then choosing computation rule CR1 (given in Figure 3.8) to solve for this goal. Conversely, the student can note that the constants given can compute, by applying CR1, the amount of alcohol in the total solution. When he compares his result to the problem, he discovers it to be the desired quantity, and reports it as the goal. His observable behavior is the application of rule CR1, which does not show how he proceeded.

In general, when a student solves problems in a goal-directed manner, his actions may be modeled by both the weak forward-directed strategy and the means-ends strategy. This is because he usually does not make his goal structure explicit, and his actions reflect the forward-directed activity of calculating values which in turn satisfy the subgoals. A similar situation in which the solution does not exhibit an apparent distinction between approaches is when compositions of rules are applied. It is impossible to tell from the problem solver’s observable behavior whether he performed the individual steps of a composition or made one composite step.

Another reason the distinction is not clear-cut is because in the model, old rules are retained at each higher level which has additional rules. Therefore, in the previous example, a student who would do goal-directed problem solving on a difficult problem may actually generate values in a weak forward-directed manner because that strategy works as well. Mayer (1982) reports that for simple problems, directly carrying out the operations is easier for students than setting up a formula. In these situations, experienced
problem solvers behave in a manner consistent with a weak forward-directed style.

Some strategic rules define behavior that is unique to a particular level of competence. If this unique behavior is observed, then evidence for concluding a student has attained that level of competence is strong. The approach suggested by this situation is akin to the concept of set difference. First, form the set of all possible problem solving states to which the strategic rules apply. Then, in each state form the set of all possible strategic rules that apply. After removing all the states where rules from different strategies apply to give the same result, you are left with states and rules that uniquely identify the strategy they come from. If more than one strategic rule applies to a particular state, then each must have a different result and the rule applied by the student can be determined from the result. If only one strategic rule applies to a particular state, then a student must apply that rule and the strategy he is employing is obvious.

Strategic rules that are compositions of component strategic rules complicate the determination of which strategy was applied. If the student goes from one state to another by performing an operation that is described by a composite rule, it is not certain that he performed the composite rule or a succession of component rules on paper before entering the result into the system. However, if the student goes through the intermediate states the component rules would go through, we know which strategy he applied.

We have not exhaustively generated and searched all such states to determine the states that discriminate between strategies in the manner described and then devised a representation to cover all such situations.

20This is another example of the bandwidth problem (Goldstein, 1979).
Instead, we have encoded rules to distinguish the strategies that we know are possible in more common situations. Evidence in the strategic rules that demonstrates the advancement from the weak forward-directed strategy to the means-ends strategy can be seen in rules ME2.1 and ME2.2, which employ a formula with two unknowns, and in ME4, which explicitly establishes subgoals.

Expert rule-selection rules employ the problem goal, and the absence of this goal in a formula provides evidence that the student has not attained the expert level but remains at the means-ends level. For example, in a protocol taken of a student solving

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

the student made the comment

".7 times x number of gals. giving (sic) the number of gallons of alcohol"

establishing a relationship with two unknowns. This fact distinguishes his processing from being weak forward-directed. Likewise the expression does not contain the problem goal, the amount of pure alcohol, and therefore the student has not applied the expert strategy, which concentrates on the problem goal.21

Strategic rule EX2.1 provides evidence that the student is doing some work at the expert level. EX2.1 employs a specialized formula to solve immediately for the problem goal. Had the student solved the problem given above with

\[ 2 + 1x = .7(10+x), \]

it would have been obvious that he applied EX2.1.

\[ ^{21}\text{Figure 3.10 gives the entire protocol and figure 3.12 simulates it on the system showing the rule that fired that concluded the means-ends strategy was used.} \]
These changes in level of competence characterize two facets of learning. One facet is the learning of new techniques demonstrated by learning to expand the use of symbols to represent problem entities. The second facet is the compiling of knowledge to improve processing. This is demonstrated by strategy rules EX2.2 and EX3, which are formed from compositions of means-ends rules.

If the model were being employed to model individual students, should it have the ability to cull out old rules that are not applied due to compiled knowledge? Anderson's (1986) simulation of learning LISP programming does not remove rules from which proceduralizations and compositions were built, because they may be applied to build new proceduralizations or compositions. In algebra word problem solving also, it is necessary to retain the more primitive rules. For example, after practicing solving other kinds of problems to the extent necessary to become expert, a student begins work with distance-rate-time problems. The expert strategies will not help him solve problems such as

A truck leaves a depot traveling north at 35 mph. Two hours later a car starts from the same depot and travels at 55 mph until it overtakes the truck. How long does it take to overtake the truck?

Let the variables $D_t$, $D_c$, $T_t$, and $T_c$ stand for distance traveled by the truck, distance traveled by the car, time of travel of the truck, and time of travel of the car respectively. Instead of applying the expert strategy the student can fall back on the means-ends strategy to establish

$$D_c = 55T_c$$

which makes $D_c$ a subgoal from which

$$D_t = D_c$$

yields $D_t = 55T_c$

which makes $D_t$ a subgoal from which
\[ D_t = 35T_t \text{ yields } 35T_t = 55T_c \]

which make \( T_t \) a subgoal from which

\[ T_t = T_c + 2 \text{ yields } 35(T_c + 2) = 55T_c. \]

This solution was arrived at solely by employing the most basic relationships and not those characterized by the expert strategy. With practice he could do the last step directly.

**Relation to the geometry tutor**

The geometry tutor (Anderson et al., 1985) employs an in-depth model of geometry problem solving in generating direct proofs of simple theorems. It also exploits the problem solving model as a primary component of a knowledge base for a tutorial system in mathematics. This work shares some of the other major ideas incorporated in al-Khorezmi, such as tracing the student's progress in a model and representing typical bugs to allow corrective feedback. However, the problem-solving tasks involved differ. In their domain, problem solving proceeds by forward and backward reasoning with rules of the form:

If \( a \) and \( b \) then conclude \( c \).

Therefore, all of their proofs can be handled by rules analogous to our strategy rules FD2.1 and ME2.1. This similarity is underscored by a common characteristic that has been noted in the protocols of both research efforts: the intermixing of forward and backward processing to find a connection in the middle. Such intermixing limits the search that would be necessary if only forward-directed or backward-directed reasoning alone were attempted.

However, because there is no concept in their domain analogous to the use of symbolic mathematics and simultaneous equations, al-Khorezmi incorporates a more diverse set of strategies by necessity and teaches
improvement through the acquisition of more powerful techniques. Learning
to do geometry proofs requires improvement by learning how to limit the
search space. Geometry proofs do not have cues to limit the search space as
obvious as those in solving algebra word problems. In algebra word problems,
problem type can serve as a primary cue, immediately limiting the operations
considered. A second way the search space can be limited is not so obvious and
involves the interplay between the weak forward-directed and means-ends
strategies. If rules with more than one unknown are considered for use with
any quantity instead of just the goal quantities, as in the means-ends strategy,
then the number of possible computation rule applications vastly increases.
We discovered this by originally modeling the weak forward-directed rules
FD2.1 and FD2.2 with the following rule:

Given an attribute with a known value, select a rule with it as
input and determine consistent entity bindings.

which appeared in Bregar et al. (1986). When implemented in the help
portion of the tutor, this rule gave numerous useless bindings to consider.
Rules FD2.1 and FD2.2 restrict the bindings to one unknown and therefore
provide useful guidance in limiting the search. Also, the more restricted rules
adequately modeled the protocol data and demonstrated the overgenerality of
the old rule.

3.2. Model enables meaningful response to student approach

The model has been constructed to model both strategies and tactics for
solving algebra word problems. Therefore, when employed to do model
tracing, the model aids recognition of the strategic operations and the tactical
operations that a student performs. With this knowledge of a student's
strategies, modeled by strategic rules, and a student's tactics, modeled by
computation rules, four approaches to responding to the student's performance are available. These are:

- *Reinforce* the lower level strategy, which employs known rules, and enlarge the student's computation rule set in other domains.

- Give the student *practice* at the current strategy with known rules.

- *Expand horizontally*, keeping the student at the same strategic level but enlarging the student's computation rule set.

- Upgrade the student's strategic repertoire by *expanding vertically*, enlarging the student's strategic rule set.

Table 3.1 shows the placement of these approaches with respect to changing strategy levels and incorporated tactical rules.

<table>
<thead>
<tr>
<th>Strategies</th>
<th>Decrease level</th>
<th>Stay at same level</th>
<th>Increase level</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keep same</td>
<td></td>
<td><em>practice</em></td>
<td><em>expand vertically</em></td>
</tr>
<tr>
<td>Add</td>
<td><em>reinforce</em></td>
<td><em>expand horizontally</em></td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1. Responses to student approach

Our strategy levels have a correlation to levels of ability. Learning the means-ends strategy enables solving problems that cannot be solved by applying the weak forward-directed strategy. Learning the expert strategy enables faster problem solving. The term *expand horizontally* indicates that we keep the level of ability constant while new computation rules are learned to enable the solving of problems from a wider range of domains. Correspondingly, the term *expand vertically* indicates an increase in ability.
Applying these four approaches to the three strategy levels in our model gives us several options for directing and guiding students. At the weak forward-directed level, the system can have the student practice what he has shown he can do, teach the student new domain rules, or tutor the student in the use of symbols in problem-solving (attempt to expand vertically). At the means-ends level, the system can reinforce the direct application of rules that apply new domain rules, have the student practice his current knowledge, teach the student new domain rules, or teach the combined strategy and domain rules that distinguish expert processing. At the expert level, the tutor is limited by the upper bound on strategy levels and cannot attempt to move up a strategy level. It can, however, reinforce the means-ends strategy as a general goal-directed problem-solving technique, provide practice opportunities, or teach new domain rules.

3.3. Implementing a tutor based on the model

We have taken the model through several steps of development and implementation that we will demonstrate in the following sections. The first step was implementation of the model as a rule-based program to demonstrate its problem solving capabilities and compare them to our protocol data. The second step was to develop a simple menu driven interface by studying how a tutor interacts with a student through a computer connection. With an interface designed, we implemented a model tracing tutor and tested its capabilities by running the interactions recorded in our student protocols through it. However, a tutoring program that is entirely reactive is inadequate when students request help. Therefore, we provided a help component. The help component determines the next step the model would take, which the tutor then gives as a hint. This component continues to give
more specific hints to the point of demonstrating the answer if the student continues to request help. Furthermore, to allow the student to explore the knowledge of al-Khorezmi, we added a passive help component. This component organizes its knowledge by employing the genetic graph, and we discuss it in the next chapter. Finally, twelve high school students exercised al-Khorezmi to test its capabilities and verify its underlying model.

3.3.1. Simulating a student by employing the model only

We recorded the protocol in Figure 3.10 while a student interacted through a computer connection with a human tutor in a separate room. Figure 3.11 gives a simulation of the operations the student performed. We employed the rule-based implementation of our problem solving model to generate the simulation. The YAPS (Allen, 1983) production system monitor, in which we implemented our model, matches its rules against a database and may have several which are ready to fire at a time. Therefore, there must be a selection technique for selecting which rule to fire. This selection technique is called conflict resolution and is provided by the production system monitor. Because the conflict resolution strategy of the YAPS production system monitor controls rule firing, the trace of the simulation does not correspond exactly to the student protocol. The important point is not that the rules fire in exactly the same order as the protocol, but that the rules applied by the student are in the conflict set at the right time. That enables the system to monitor the steps of a student problem solver; thus the firing order of YAPS is not significant.
The problem you are to solve is as follows:
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

S.1 I have a 20% solution and I must add another complete solution to it to obtain the 70% solution. Is it important to know how much in gallons is in the 70% solution? Because the tutor cannot see the student's work in this protocol, the bandwidth problem (Goldstein, 1979), he concluded that the student established a subgoal, the amount of the 70% solution, from relationship: the amount of the 70% solution = 10 + the amount of pure alcohol.

T.1 Yes, although that is not what you are to find to answer the problem, it certainly can be used to find the answer.
The tutor supports the selected subgoal.

S.2 I have twenty percent or two gallons of alcohol in the first solution. I have 100% alcohol in the second solution, and 70% in the third. 100% equals one. Help!

Here the student generated a quantity in a weak forward-directed manner without any idea of future usage.

T.2 You told me you have 2 gallons of alcohol in the first solution. You also told me you have 100% alcohol in the second solution. How many gallons of alcohol do you have in the second solution?

S.3 If I have 70% sol. Then I have .7 times X number of gals. giving me the number of gallons of alcohol. I know there are more than two, there are 8 gals. of water total. 30% equals 8 gals. Now I have 3 times X number of gallons giving me 8 gallons. So 8 divided by .3 should give me the total solution.

Here the student established a subgoal to find the number of gallons of alcohol in the 70% solution from: the number of gallons of the 70% solutions is .7 times the number of gallons of alcohol in the 70% solution. He then switched to a weak forward-directed strategy to generate the amount of water in the 70% solution as 8 and the proportion of water in the 70% solution as .3. This does not produce the amount of alcohol in the 70% solution, but rather produces the amount of the 70% solution, the supergoal for finding the amount of alcohol in the 70% solution.

T.3 Which is how many gallons?

S.4 26.6 gals.

T.4 Is that the answer to the problem or would you like to continue? To remind you, the problem is, A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution.
The tutor has followed the student's work even though the student's solution technique is very different from the expert technique the tutor would have employed.

S.5 I want to go on. The problem (sic) I just gave was only the total number of gallons in the third solution. Now I can multiply .7 times 26.6 giving me 18.62. That is the amount of pure alcohol 18.62 gals.

Here the student computed the subgoal, amount of alcohol in the 70% solution, even though he already found its supergoal.

Figure 3.10. Computer link student-tutor protocol
T.5 Gallons of pure alcohol in which solution?

S.6 I forgot to subtract the two from the first solution. This would make the answer 16.62 gals. of pure alcohol. The student applied a weak forward-directed strategy to compute the problem goal from the amount of alcohol in the 70% solution instead of exploiting the satisfied subgoal, the amount of the 70% solution, to compute the problem goal.

T.6 Correct, you did a great job.

When simulating this interaction with al-Khorezmi, the values that are in **bold** characters are those that were entered into al-Khorezmi as constants, variables, and calculated results and the segments that are *underlined* are those that were entered into al-Khorezmi as equations. The comments we have added to this protocol are in *italics*.

Figure 3.10 (continued)
The problem to be solved is as follows:

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

The primary goal in this problem is amount-value in the pure solution (x00037).

Beginning consideration of rules in the means-ends strategy.

A rule has been instantiated to solve amount-value in the pure-solution (x00037).  ME2.1.  The rule is cr2.  x00037 = (plus x00040 -10.0)

The subgoal amount-value in the 70%-solution has been set up.  ME4.  These beginning steps correspond to S.1 and T.1 where the student determines she must look for the gallons of the 70% solution.

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: cr3  FD2.1  x00009 = (times 0.2 10.0)

Will now execute the computation rule.  FD3.1.  cr3 x00009 = (times 0.2 10.0)  Variable x00009 replaced with 2.0 - amount-value in alcohol-20%-solution incorporated as 2.0  This corresponds to the forward generation of the amount two gallons in S.2.

Beginning consideration of rules in the means-ends strategy.

A rule has been instantiated to solve amount-value in the 70%-solution (x00040).  ME2.2.  The rule is cr3.  x00027 = (times 0.7 x00040)

This corresponds to the relationship expressed in S.3 for computing the gallons of the 70% solution.

The subgoal amount-value in alcohol-70%-solution has been set up.  ME4.

The following set of actions are applied to compute the 8 gallons of water total mentioned in S.3.

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: cr10  FD2.2  10.0 = (plus 2.0 x00014)

Will now solve the equation: cr10  FD3.2  10.0 = (plus 2.0 x00014)  Variable x00014 replaced with 8.0 - amount-value in water-20%-solution incorporated as 8.0

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: cr7  FD2.1  x00013 = (plus 1.0 -0.2)

Will now execute the computation rule: cr7  FD3.1  x00013 = (plus 1.0 -0.2)  Variable x00013 replaced with 0.8 - proportion-value in water-20%-solution incorporated as 0.8

Figure 3.11. Simulation of student actions in Figure 3.10
Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( cr_7 \ FD_{2.1} \ x_{00022} = (\text{plus} \ 1.0 \ -1.0) \)

Will now execute the computation rule: \( cr_7 \ FD_{3.1} \ x_{00022} = (\text{plus} \ 1.0 \ -1.0) \)

Variable \( x_{00022} \) replaced with 0.0 - proportion-value in water-pure-solution incorporated as 0.0

Beginning consideration of rules in the weak forward-directed strategy.

In \( cr_3, \ FD_{2.1} \) 0.0 is filled in for the amount of the 0.0 proportion water-pure-solution. \( FD_{3.1} \)

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( cr_8 \ FD_{2.1} \ x_{00032} = (\text{plus} \ 8.0 \ 0.0) \)

Will now execute the computation rule: \( cr_8 \ FD_{3.1} \ x_{00032} = (\text{plus} \ 8.0 \ 0.0) \)

Variable \( x_{00032} \) replaced with 8.0 - amount-value in water-70%-solution incorporated as 8.0

The 8 gallons of S.3 has been determined.

The following determines the proportion .3 employed in S.3.
Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( cr_7 \ FD_{2.1} \ x_{00031} = (\text{plus} \ 1.0 \ -0.7) \)

Will now execute the computation rule: \( cr_7 \ FD_{3.1} \ x_{00031} = (\text{plus} \ 1.0 \ -0.7) \)

Variable \( x_{00031} \) replaced with 0.3 - proportion-value in water-70%-solution incorporated as 0.3

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( cr_3 \ FD_{2.2} \ 8.0 = (\text{times} \ 0.3 \ x_{00040}) \)

This is the formula established in S.3, which employs the previously generated 8 and .3.

Will now solve the equation: \( cr_3 \ FD_{3.2} \ 8.0 = (\text{times} \ 0.3 \ x_{00040}) \)

Variable \( x_{00040} \) replaced with 26.67

This is the result given in S.5.

\( cr_3 - x_{00027} = (\text{times} \ 0.7 \ 26.67) \)

Will now execute the computation rule: \( cr_3 \ FD_{3.1} \ x_{00027} = (\text{times} \ 0.7 \ 26.67) \)

Variable \( x_{00027} \) replaced with 18.67 - amount-value in alcohol-70%-solution incorporated as 18.67

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( cr_8 \ FD_{2.2} \ 18.67 = (\text{plus} \ 2.0 \ x_{00018}) \)

Figure 3.11 (continued)
Will now solve the equation: \( \text{cr8 } FD3.2 \)
\[18.67 = (\text{plus } 2.0 \times 00018)\]
Variable \( x00018 \) replaced with 16.67 - amount-value in alcohol-pure-solution incorporated as 16.67
This computes the answer given in S.6.

Beginning consideration of rules in the weak forward-directed strategy.

A value can be directly computed from: \( \text{cr3 } FD2.2 \)
\[16.67 = (\text{times } 1.0 \times 00037)\]

Will now solve the equation: \( \text{cr3 } FD3.2 \)
\[16.67 = (\text{times } 1.0 \times 00037)\]
Variable \( x00037 \) replaced with 16.67
\[\text{cr2 } - 16.67 = (\text{plus } 26.67 -10.0)\]

Goal of finding amount-value in the 70%-solution satisfied with 26.67 \( MEI \) Goal was incorporated in \( \text{cr2 } 16.67 = (\text{plus } 26.67 -10.0) \)

amount-value in the pure-solution incorporated as 16.67

Goal of finding amount-value in the pure-solution satisfied with 16.67 \( FD1 \)
This reports the answer given in S.6.

Strategy rule names have been added in italics because they were not part of the original output. Because our terminology (strategy names and rule names) has changed since we ran this simulation, we have updated the output to reflect this change. Also, the original problem solver employed gensyms to name the problem entities. We replaced them here with the names employed by al-Khorezmi to improve the readability of the trace.

Figure 3.11 (continued)

3.3.2. Providing a preliminary interface for model tracing

Students and tutors pass a significant amount of information back and forth during a tutorial session. Therefore, an effective intelligent computer-assisted-instruction word problem tutor needs to do the same. Enabling this information transfer in a preliminary interface while testing other more significant facets of the tutor caused us concern. Therefore, we developed the interface with two goals in mind:
• allow the communication of the necessary information, and
• keep the implementation as simple as possible to allow concentration on
  the model tracing and error correction.

We attacked the first goal by examining student-tutor protocols that
were taken online during tutoring through a computer connection between
terminals in separate rooms. We noted that there are many ways to refer to
the various quantities and entities in the problem statement. Indeed, the tutor
often had trouble getting the student to understand exactly which quantity or
entity he was referring to. Likewise, the student had trouble expressing the
quantity to which he was referring. Therefore, it would be very difficult to
implement these references to problem entities in a natural language
interface. We decided to have al-Khorezmi continually display descriptive
names for these quantities as options in a menu.

In the protocols, we noted another significant facet of the interactions.
Students, especially the poorer ones, often had trouble recalling the quantities
of the problem that had already been noted or solved for. Therefore, to
provide the student with a memory aid, we decided to display the entity
schemas and continuously update them with values for quantities as they
became established.

Finally, the operations taken by the student need to be specified to al-
Khorezmi. Examination of the protocols uncovered five operations that were
applied in 46 of 65 student interactions or 71% of the time. We encoded these
operations into menu options.
1) note goal, 
2) note constant, 
3) specify variable, 
4) give calculated result, and 
5) give expression. 

We added the option, 6) solve, since it was something the student did in the background and something al-Khorezmi can do upon request. Five of the interactions were requests for help and 13 contained natural language statements or questions, some of which would not have been necessary if all the information in this interface had been in front of the student at the time he was working on the problem. In response to the request for help, the tutors typically provided hints in the form of a leading question about what the next step in solving the problem should be. This type of response will be discussed in the section below on hints.

3.3.3. Testing the interface with student protocols

For the interface to meet the goal of communicating the necessary information to the tutor, it must accept and understand at least the information provided by students in the recorded protocols. Also, it must perform in a manner that results in a beneficial interaction with students in a live tutorial session. Figure 3.12 shows a run of the tutor with the student operations from the protocol in Figure 3.10. We present this protocol because it shows considerable variability in the use of strategies and domain rules. Therefore, it provides a good test of the capabilities of our interface and of our model.

To test our interface with our protocols we devised the following set of rules to make the encoding of protocols into the interface consistent. These rules are as follows:
1) When a student makes a statement indicating he has a goal, note it.

2) When a constant, variable, or result is mentioned, enter it.

3) When a formula is stated, first enter all quantities in the formula by applying rule (2) above. After the quantities are entered, enter the formula by employing the menu options (any quantities mentioned by name are referred to by menu option).

4) And, when a result is given for a formula, apply the solve option to make al-Khorezmi derive the same result.

The protocol in Figure 3.10 has been marked to indicate the items that matched these rules so that the reader can easily make the correspondence between the protocol and the program run. The constants, variables, and results mentioned in rules two and four have been printed in boldface type. Finally, the formulas mentioned by rule three have been underscored.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

<table>
<thead>
<tr>
<th>Amount</th>
<th>Proportion</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alcohol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Water</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a) amount-20%-solution
b) amount
c) proportion
d) percent
e) amount
f) proportion
g) percent
h) amount-pure-solution
i) amount
j) proportion
k) percent
l) amount
m) proportion
n) percent

o) amount-70%-solution
p) amount
q) proportion
r) percent
s) amount
t) proportion
u) percent
v) full proportion constant 1.0
w) full percent constant 100.0
y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result. Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

Option? 2 Quantity? d Value? 20
OPERATION VERIFIED TO BE CORRECT

Option? 2 Quantity? r Value? 70
OPERATION VERIFIED TO BE CORRECT

Option? 4 Quantity? b Value? 2
OPERATION VERIFIED TO BE CORRECT

Figure 3.12. Running a protocol through the interface
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

<table>
<thead>
<tr>
<th>Alcohol</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) amount-20%-solution 10.0</td>
<td>b) amount 2</td>
</tr>
<tr>
<td>c) proportion 0.2</td>
<td>d) percent 20</td>
</tr>
<tr>
<td>e) amount</td>
<td>f) proportion</td>
</tr>
<tr>
<td>g) percent</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alcohol</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>h) amount-pure-solution</td>
<td>i) amount</td>
</tr>
<tr>
<td>j) proportion 1</td>
<td>k) percent 100</td>
</tr>
<tr>
<td>m) proportion</td>
<td>n) percent</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Alcohol</th>
<th>Water</th>
</tr>
</thead>
<tbody>
<tr>
<td>o) amount-70%-solution x1</td>
<td>p) amount</td>
</tr>
<tr>
<td>q) proportion 0.7</td>
<td>r) percent 70</td>
</tr>
<tr>
<td>s) amount</td>
<td>t) proportion</td>
</tr>
<tr>
<td>u) percent</td>
<td></td>
</tr>
</tbody>
</table>

v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) helpHints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 5 Expression? q*o=p Hit return to continue.

Menu option p is as yet unspecified. Please enter a value or variable for it: x 2

Running rule expression-does-not-contain-primary-goal

Facts used:
- 279. (process strategy-determination)
- 262. (strategy-believed-to-be-in-effect fd-selections)
- 264. (goal <quantity-description f100>)
- 284. (schema-variables 283 x01036 x1)
- 285. (schema-variables 284 x01031 x2)

Removing fact from db <yaps-database f0>:
- 262. (strategy-believed-to-be-in-effect fd-selections)

Adding fact to db <yaps-database f0>:
- 290. (strategy-believed-to-be-in-effect me-selections)

Adding fact to db <yaps-database f0>:
- 291. (strategy-prediction me-selections)

EXPRESSION VERIFIED TO BE CORRECT
OPERATION VERIFIED TO BE CORRECT

(0.7 * x1) = x2

Option? 4 Quantity? s Value? 8
OPERATION VERIFIED TO BE CORRECT
(0.7 * x1) = x2

Figure 3.12 (continued)
Option? 4  Quantity? u  Value? 30
OPERATION VERIFIED TO BE CORRECT
(0.7 * x1) = x2

Option? 4  Quantity? t  Value? .3
OPERATION VERIFIED TO BE CORRECT
(0.7 * x1) = x2

Option? 5  Expression? t*o=s  Hit return to continue.

EXPRESSION VERIFIED TO BE CORRECT
(0.7 * x1) = x2
(0.3 * x1) = 8

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem

Enter a list of numbers in increasing order designating the expressions you want to solve - (2)

Would you like to 1) compute the single unknown  2) isolate the unknown goal on the left 1

Variable x1 replaced with 26.67
(0.7 * 26.67) = x2

Figure 3.12 (continued)
Option? 6

Would you like to 1) compute the single unknown 2) isolate the unknown goal on the left 

Variable x2 replaced with 18.67

Option? 5 Expression? p-b=h Hit return to continue.

Menu option h is as yet unspecified. Please enter a value or variable for it:x3

EXPRESSION VERIFIED TO BE CORRECT
OPERATION VERIFIED TO BE CORRECT

(18.67 + -2) = x3

Option? 6

Would you like to 1) compute the single unknown 2) isolate the unknown goal on the left 

Variable x3 replaced with 16.67

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

<table>
<thead>
<tr>
<th>alcohol</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) amount-20%-solution 10.0</td>
<td>b) amount 2</td>
</tr>
<tr>
<td>c) proportion 0.2</td>
<td>d) percent 20</td>
</tr>
<tr>
<td>e) amount</td>
<td>f) proportion</td>
</tr>
<tr>
<td>g) percent</td>
<td></td>
</tr>
<tr>
<td>h) amount-pure-solution 16.67</td>
<td></td>
</tr>
<tr>
<td>i) amount</td>
<td>j) proportion 1</td>
</tr>
<tr>
<td>k) percent 100</td>
<td></td>
</tr>
<tr>
<td>l) amount</td>
<td>m) proportion</td>
</tr>
<tr>
<td>n) percent</td>
<td></td>
</tr>
<tr>
<td>o) amount-70%-solution 26.67</td>
<td></td>
</tr>
<tr>
<td>p) amount 18.67</td>
<td>q) proportion 0.7</td>
</tr>
<tr>
<td>r) percent 70</td>
<td></td>
</tr>
<tr>
<td>s) amount 8</td>
<td>t) proportion 0.3</td>
</tr>
<tr>
<td>u) percent 30</td>
<td></td>
</tr>
<tr>
<td>v) full proportion constant 1.0</td>
<td>w) full percent constant 100.0</td>
</tr>
<tr>
<td>x) other value</td>
<td></td>
</tr>
</tbody>
</table>

1) note goal 5) give expression 9) help-algebra
2) note constant 6) solve 10) exit-problem
3) specify a variable 7) help-hints
4) give calculated result 8) help-menu-options

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason ?

Figure 3.12 (continued)

3.3.4. Employing the model to provide hints

We left the design of a full tutorial component until later so that more attention could be given to pedagogy by having available the completed
knowledge representation scheme. In the meantime, we examined the protocols for typical interactions that could be implemented. We found that the tutor usually asked a question formulated from what the next step in solving the problem should be. In 50 of 59 tutorial interactions, the tutor asked such questions. Because it is built around a problem-solving model itself, al-Khorezmi allowed easy implementation of this tutorial strategy. Al-Khorezmi makes a determination of the strategy it believes the student is applying in the manner described previously, and then gives a hint based on the next step al-Khorezmi would take under that strategy. If the student requires more help, then al-Khorezmi repeatedly gives more specific hints upon each request until it provides the student with what it is hinting at. This process can be repeated continually until the answer to the problem is reached. Figure 3.13 gives an example of this hint giving facility.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

<table>
<thead>
<tr>
<th>alcohol</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) amount-20%-solution 10</td>
<td>b) amount c) proportion 0.2 d) percent 20</td>
</tr>
<tr>
<td>e) amount f) proportion g) percent</td>
<td></td>
</tr>
</tbody>
</table>

h) amount-pure-solution

<table>
<thead>
<tr>
<th>alcohol</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>i) amount j) proportion 1 k) percent 100</td>
<td></td>
</tr>
<tr>
<td>l) amount m) proportion n) percent</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>alcohol</th>
<th>water</th>
</tr>
</thead>
<tbody>
<tr>
<td>p) amount q) proportion 0.7 r) percent 70</td>
<td></td>
</tr>
<tr>
<td>s) amount t) proportion u) percent</td>
<td></td>
</tr>
</tbody>
</table>

v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal  5) give expression  9) help-algebra
2) note constant 6) solve  10) exit-problem
3) specify a variable 7) help-hints
4) give calculated result 8) help-menu-options

Try to use the strategy rule: if THERE IS A RULE THAT USES AS INPUT A PRIMARY GOAL OR A SUBGOAL then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS; to find a rule that will find amount-pure-solution

Choose help-hints again for a more detailed hint.

The computation rule: THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION; can be used to find amount-pure-solution

Choose help-hints again for a more detailed hint.

The goal amount-pure-solution can be found from: amount-alcohol-pure-sol( ) = proportion-alcohol-pure-sol( ) * amount-pure-solution( )

Choose help-hints again for a more detailed hint.

The goal amount-pure-solution can be found from: amount-alcohol-pure-sol(Unknown) = proportion-alcohol-pure-sol(1) * amount-pure-solution(Unknown)

---

Figure 3.13. Giving hints based on the problem-solving model
Choose help-hints again for an alternate approach.

Option? 7

Try to use the strategy rule: if THERE IS A RULE THAT USES AS INPUT A PRIMARY GOAL OR A SUBGOAL then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS; to find a rule that will find amount-pure-solution

Choose help-hints again for a more detailed hint.

Option? 7

The computation rule: WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF THE COMPOSITE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE CONSTITUENT SOLUTIONS; can be used to find amount-pure-solution

Choose help-hints again for a more detailed hint.

Option? 7

The goal amount-pure-solution can be found from: amount-70%-solution( ) = amount-20%-solution( ) + amount-pure-solution( )

Choose help-hints again for a more detailed hint.

Option? 7

The goal amount-pure-solution can be found from: amount-70%-solution(Unknown) = amount-20%-solution(10) + amount-pure-solution(Unknown)

Choose help-hints again for an alternate approach.

Figure 3.13 (continued)

3.3.5. Verification of the model by experimental tutoring

To test our hypothesis that our problem-solving model will fill the role of the central component of the knowledge base for a tutorial system in mathematics, we had twelve students solve problems presented by al-Khorezmi. The students were from Lutheran High School in Portland, Oregon, and were excused from one 42-minute math class to take part in the
experiment. We saved their interaction with al-Khorezmi by running the it under a shell within the EMACS editor.22

Demonstrating strategic model usage

The students worked all or part of 33 problems. The weak forward-directed, means-ends, and expert strategies modeled all of their actions, except for a brief attempt at hill-climbing by one student. The one student who tried hill-climbing for a moment, apparently saw that the tutor did not support her efforts and applied the expert strategy. Besides the hill-climbing attempt, al-Khorezmi matched all of the strategic operations performed by the students to strategic rules. Al-Khorezmi recorded use of each of the strategic rules except ME1, ME3.2, and ME4. However, on two occasions students established an equation in the goal and one other unknown and became stuck. At this point, making the other unknown a subgoal (ME4) makes all the relationships pertaining to that quantity applicable. Because more relationships are applicable, a student with limited knowledge would be more likely to know of one that would allow him to proceed. These two students were working on paper because they worked faster than al-Khorezmi could respond. The

---

22The tutor operates quite slowly on a VAX 750 running Unix. For example, firing three YAPS rules to verify a constant is correct can involve some waiting. This is at least in part due to YAPS not applying its pattern matching to the Flavors objects it allows to be used for storing structured representations of objects. The tutor is especially slow when presented with complex formulas. Therefore, on occasion we encouraged the students to go ahead and work on paper. For this purpose we provided paper printed with the menu of problem entities the program displays for the student so that the students could perform program operations while waiting for the system to respond. On the first day of experimental usage we discovered that loading a new problem into the tutor took too much time given the time constraints mentioned above. Therefore, on the following days we provided different copies of the tutor, each loaded with a different problem for the students to run. This prevented the tutor from maintaining its record of the rules it observed the students using. The system has been rerun to allow the individual problems to be loaded while entering the complete interactions recorded during the experiment by the system and on paper. The output of these runs of the system appears in Appendix B.
remaining time in the period was limited so the experimenter chose to suggest another expression in their two unknowns as strategy rule EX3.2 would direct them to do.

Demonstrating computation rule usage

Of the 33 problems worked on during the experimental usage of al-Khorezmi, all the correct applications of computation rules by the students were recognized except one. The one application of a correct rule that was not recognized was on the problem

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

Student LE established the correct equation

\[
\frac{7}{3} \cdot \frac{x}{8}
\]

which is a proportion. Chapter 2 discussed our avoidance of obtaining solutions by setting up proportions, because we are interested in general techniques applicable to domains where proportions are not possible. Had the student applied the equivalent formula

\[
\frac{8}{.3} = \frac{x}{.7}
\]

al-Khorezmi would have recognized it as a valid combination of rules it knows.

Demonstrating strategy determination

Section 3.1.4 discusses how the level of a student can be determined from comparison of his actions to the strategic rules. The protocol in Figure 3.12 shows al-Khorezmi determining that the student is applying the means-ends strategy, because he is applying a rule that has two unknowns but does not include the problem goal. Student TW solved easy problems in a manner
consistent with the weak forward-directed strategy; when presented with difficult problems, student TW solved them with an expert strategy. Figure 3.14 presents a segment of student TW's interaction with al-Khorezmi. In this segment, al-Khorezmi determines that student TW is applying the expert level strategy. The determination is made by noting that a complex, specialized computation rule in only the goal unknown was employed (strategy rule EX2.1). Appendix B contains the entire recordings of the students' interactions. The recording of TW's interactions shown in the appendix further demonstrates his expertise in solving these problems.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>acid</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a) amount-30%-solution 10.0
b) amount 3
c) proportion 0.3
d) percent
e) amount 7
f) proportion 0.7
g) percent

h) amount-15%-solution 20.0
i) amount 3
j) proportion 0.15
k) percent
l) amount 17
m) proportion 0.85
n) percent

o) amount-mixture 30
p) amount 6
q) proportion **
r) percent
s) amount
t) proportion
u) percent

v) full proportion constant 1.0
w) full percent constant 100.0
y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 5
Expression? c*a+j*h=x1*o Hit return to continue.

Which quantity is variable x1 in: \((0.3 \times 10.0) + (0.15 \times 20.0)) = (x1 \times 30)\) q

Running rule student-used-specialized-rule-with-single-goal-unknown

Facts used:
594. (process strategy-determination)
205. (strategy-believed-to-be-in-effect fd-selections)
273. (goal <quantity-description f82>)
601. (student-expression x02286 equate (x02287 nil) (x02294 nil))
599. (student-expression x02287 plus (x02288 nil) (x02291 nil))
600. (student-expression x02294 times (x1 (composite-solution first-substance proportion-value)) (30 (composite-solution amount-value)))
596. (schema-variables 595 x01067 x1)

Removing fact from db <yaps-database f0>:
205. (strategy-believed-to-be-in-effect fd-selections)

Adding fact to db <yaps-database f0>:
603. (strategy-believed-to-be-in-effect ex-selections)

Adding fact to db <yaps-database f0>:
604. (strategy-prediction ex-selections)

Figure 3.14. Determining use of the expert strategy
Demonstrating compiled knowledge

As we discuss earlier, our model supports the conclusion of Neves and Anderson (1981) that learners improve their ability through linear composition of two or more rules into a single rule. Also, we show that one form of compiled knowledge in mathematical relationships is the replacement of terms in an expression with subexpressions to form the hierarchical combination of expressions. Our experimental usage of the tutor (recorded in appendix B) has demonstrations of these types of knowledge compilations. Student TW exhibits the hierarchical combinations of rules in the form of the specialized computation rules that are applied by EX2.1. Student CP, a calculus student, demonstrated the use of strategic rules EX2.2 and EX3 which, as noted previously, are formed from compositions of means-ends rules. Student CW started a problem with values for quantities that can be obtained only after four applications of FD2.1 and FD3.1. Counting these four applications of FD2.1 and FD3.1 does not consider the operation of converting percent to proportion. Recall that the composed application of FD2.1 and FD3.1 is the form of compiled knowledge we call spontaneous computation. In addition, other less obvious demonstrations of compiled knowledge appear in the interactions in the appendix.
4. The Genetic Graph

4.1. The importance of the genetic graph

The previous chapter details an in-depth model of problem solving in the domain of algebra word problems, and points out the tutorial leverage gained by use of the model. However, the model by itself is simply a group of discrete points in rule space with no interconnections between them. The genetic graph (Goldstein, 1979) provides meaningful interconnections between the rules for use in tutoring. Chapter 2 demonstrates that the relationships established by the genetic graph are those relationships considered important to learning. Therefore, a student has a better chance of mastering a rule when presented with the rule and its genetic relationship to knowledge the student holds.

In addition to our first thesis:

A tutorial system in mathematics must have as a central component of its knowledge base, a problem solver that performs as a student does, both correctly and incorrectly.

which we discuss in Chapter 3, our second thesis about knowledge representation for intelligent tutoring is:

Supplementing the model with relationships that provide genetic information will improve the quality and value of the explanations given by the system.

In particular, we argue that this genetic information captures processes related to learning.

Goldstein (1979) attributes the importance of the genetic graph to four points:
1) The graph can provide a model of the student's knowledge by showing the connected rules that the student knows and therefore, the topology of his knowledge.

2) The graph can be employed to explain bugs in students' performance, represented as deviant rules, by relating them to the correct rules with deviation links.

3) The graph relates new knowledge to old knowledge in a meaningful way during explanation.

4) The graph provides a syllabus from which to choose a next topic based on the student's current knowledge. This topic can be chosen from the frontier of the student's knowledge—that is, the unknown knowledge directly connected to the knowledge he holds.

We have extended the graph with relationships, such as the combination, composition, and derivation relationships defined in Chapter 2, that express sophistication or difficulty. These relationships are important because they explain how new knowledge and abilities correspond to old knowledge and abilities. Because combination and composition operations combine rules to form new rules, they provide the explanation for the construction of new rules. Similarly, derivation explains the construction of new rules through algebraic manipulation.

4.2. Domain rule organization

In the following sections, we demonstrate how genetic links relate domain rules in the context of wet-mixture problems. The genetic graph organizes other domains, too. We will draw on this organization to demonstrate the concept of interdomain analogies.

4.2.1. Specialization/generalization

The specialization/generalization relationship is the relationship between an instantiated rule and an uninstantiated one. Generally a rule has far too many specializations to be explicitly represented in a knowledge base. In the
classroom, teachers teach the generalized rule and demonstrate the rule with specializations. In algebra word problem solution, generalized rules are specialized to the problem at hand. For example, given computation rule 1 (CR1) and problem 1 in Figure 4.1, CR1 specializes to the computation $12 + 10$, which gives the output, the amount of the total solution.

Problem 1:

If you mix 12 gallons of a 10% alcohol solution with 10 gallons of pure alcohol, how many gallons will be in the total solution?

Problem 2:

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

CR1: The amount of a solution is equal to the sum of the amounts of its parts.

<table>
<thead>
<tr>
<th>Structure:</th>
<th>Composite solution C from solutions S1 and S2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constraints:</td>
<td>$S1 \not= S2$</td>
</tr>
<tr>
<td>Input values:</td>
<td>amount($S1$), amount($S2$)</td>
</tr>
<tr>
<td>Output value:</td>
<td>amount($C$)</td>
</tr>
<tr>
<td>Computation:</td>
<td>$\text{amount}(S1) + \text{amount}(S2)$</td>
</tr>
</tbody>
</table>

Figure 4.1. Problem 1, Problem 2, and CR1 for specialization

Because word problems are stated with real number constants and there are an infinite number of real numbers, when the constants are varied in a given problem there are an infinite number of possible instantiations (specializations) of computation rules. Also, significant subclasses of computation rule instantiations exist for problems that have quantities with values one or zero employed respectively as the multiplicative and additive identities in the applicable computation rules. Another significant subclass of computation rule instantiations appears when the value of a quantity is zero and the applicable rule multiplies the zero times another quantity. For example,
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution? has the relationships:

- amount of alcohol in pure solution = 1 times amount of pure solution (multiplicative identity subclass)

- amount of water in pure solution = 0 times amount of pure solution (zero multiplication subclass), and

- amount of water in the 20% solution + 0 = amount of water in the 70% solution (additive identity subclass)

among others. These subclasses of rule instantiations define important specialized rules (proceduralization in Anderson's terminology) that are applied in problem solving.

Because there are infinitely many problems to which a computation rule applies, there are infinitely many specializations of each computation rule, and it is impossible to represent every specialization. Instead, we represent the generalized rule by employing variables that are unified with problem constants by the rule interpreter to form the specializations of the rule. These specializations expand the genetic graph dynamically to include specialized rules applicable to the problems being solved.

4.2.2. Analogies between specializations

When applied to problem 2 in Figure 4.1, CR1 has its computation and output value slots specialized as follows:

**Computation:** 10 + unknown amount of pure alcohol.

**Output value:** the amount of the 70% alcohol solution.

This specialization of CR1 is by definition analogous to the specialization given in the previous section of CR1 applied to problem 1 of Figure 4.1. If a student,

23Expanding the genetic graph dynamically was an uncompleted goal of the Wusor project (Goldstein, 1979).
who is experienced at performing simpler specializations with constant values (such as on problem 1 of Figure 4.1), is shown that the process is identical to specialization with unknowns, then he may find it easier to deal with unknowns. Furthermore, the recording and succeeding review of specializations performed during problem solving can reinforce the experience gained through practice.

4.2.3. **Refinement/simplification**

The refinement of a rule adds one or more conditions to the condition part of the rule. This constrains the rule to apply to a subset of the situations to which the original rule applied. The diagram in Figure 4.2 shows the refinements and associated analogies (next section) in the computation rules of the wet-mixture problem domain. In Figure 4.2 we place the rule names in a border and link them with lines that are labelled either Refinement/Simplification or Analogy depending on which relationship holds between the rules. For example, SumAmounts is linked to each of CR1, CR8, and CR10, because each of these is a refinement of SumAmounts (and SumAmounts is a simplification of each of them).
Figure 4.2. Graph of refinements and analogies for wet-mixture problems.
We employ the simplified computation rule:

**SumAmounts**: The amount of an entity is equal to the sum of the amounts of its parts.

to demonstrate refinements. There are three refinements of SumAmounts, CR1, CR8, and CR10. We state these rules as follows:

**CR1**: The amount of a composite solution is the sum of the amounts of the two participating solutions.

**CR8**: The amount of a substance in the composite solution is the sum of the amounts of the like substances in the participating solutions.
CR10: The amount of a solution is the sum of the amounts of the two participating substances.

In Figure 4.3 we give "structured" English statements of these rules. Examine the structure slots to see how the refinement of SumAmounts occurs.

| SumAmounts: | The amount of an entity is equal to the sum of the amounts of its parts. |
| Constraints: | entity E composed of parts P1 and P2 |
| Constraints: | P1 not equal P2 |
| Input values: | amount(P1), amount(P2) |
| Output value: | amount(E) |
| Computation: | amount(P1) + amount(P2) |

CR1: The amount of a solution is equal to the sum of the amounts of its parts.

| Constraints: | Composite solution C from solutions S1 and S2 |
| Constraints: | S1 not equal S2 |
| Input values: | amount(S1), amount(S2) |
| Output value: | amount(C) |
| Computation: | amount(S1) + amount(S2) |

CR8: The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating substances.

| Structure: | Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 |
| Constraints: | C1 not equal C2 |
| Input values: | amount(S2), amount(S3) |
| Output value: | amount(S1) |
| Computation: | amount(S2) + amount(S3) |

CR10: The amount of a solution is equal to the sum of the amounts of the two participating substances.

| Structure: | Solution S from substances S1 and S2 |
| Constraints: | S1 not equal S2 |
| Input values: | amount(S1), amount(S2) |
| Output value: | amount(S) |
| Computation: | amount(S1) + amount(S2) |

Figure 4.3 Structured English statements of CR1, CR8, and CR10.

4.2.4. Analogies between refinements

Note that rules CR1, CR8, and CR10 above are in fact analogous, i.e., they all sum the amount of two components of a mixture to get the total amount of
the mixture. Therefore, we link these rules with analogy links. These analogy links connect rules that have some conditions the same, because they are refined from the same simplified rule and there is a mapping between their other conditions. The analogies discussed in section 4.2.2 differ by making a mapping between the different specializations of the variables in a rule. Goldstein (1979) did not define analogies between refinements; he limited his definition to analogous specializations. Figure 4.2 shows the analogies of this type in the wet-mixture problem domain.

4.2.5. Interdomain analogies

In Figure 4.4 we give CR3 and the simplified-proportion rule from which CR3 is refined. Other domains apply the simplified-proportion rule, and it has different refinements in those domains. For example, in profit and discount problems (Mayer, 1981) the rule incorporates monetary amounts. In the discount-simplified-proportion rule given in Figure 4.4, the structure is constrained to a price P and discount D in comparison with the solution S and substance P of CR3.
**Simplified-proportion:** When you have a proportion that applies to a certain whole, multiplying the proportion times the whole amount gives an amount of a part of the whole.

- **Structure:** Whole quantity Q which may have part P
- **Constraints:** none
- **Input values:** amount(Q), proportion(P)
- **Output value:** amount(P)
- **Computation:** proportion(P) * amount(Q)

**Discount-simplified-proportion:** The amount of discount on a full price can be found by multiplying the proportion of the discount times the amount of the full price.

- **Structure:** Full price P with discount D
- **Constraints:** none
- **Input values:** amount(P), proportion(D)
- **Output value:** amount(D)
- **Computation:** proportion(D) * amount(P)

**CR3:** The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.

- **Structure:** Solution S with substance P
- **Constraints:** none
- **Input values:** amount(S), proportion(P)
- **Output value:** amount(P)
- **Computation:** proportion(P) * amount(S)

**Times:** The arithmetic operator times is employed when things are repeated, taken a proportion of, or done at a rate.

- **Structure:** none
- **Constraints:** none
- **Input Values:** Quantity1, Quantity2
- **Output Value:** Quantity3
- **Computation:** Quantity1 * Quantity2

**Simplified-rate:** A rate (other than interest) times a time gives the quantity that was measured in units per time.

- **Structure:** Quantity Q and associated rate and time.
- **Constraints:** numerator(units(rate(Q))) equals units(Q)
- **Input Values:** rate(Q), time(Q)
- **Output Value:** Quantity(Q)
- **Computation:** rate(Q) * time(Q)

---

Figure 4.4. Structured English computation rules for demonstrating refinement analogies

Weaker analogies branch out from a more distant branch in the graph of refinements and analogies given in Figure 4.2. For example,
distance = rate \times \text{time}

the rule of motion and current problems is a refinement of the simplified-rate rule stated in Figure 4.4. The simplified-rate rule is a refinement of the times rule stated in Figure 4.4 and shown in the graph of Figure 4.2. Therefore, DRT is weakly analogous to CR3 of wet-mixture problems.

4.2.6. \textbf{Derivation/origination}

Derivation links connect basic knowledge to knowledge derived from it. We define \textit{base} rules as those that reflect the most "familiar" form of a computation. For example, it is more reasonable to think of the whole being the sum of the parts than to think of a part as being the whole minus the other part. A \textit{derived} rule is said to be derived from a base rule if it binds its variables to the same entities as the base rule, the quantity it computes is an "input" to the base rule, and algebraically it is an equivalent expression. If FD2.2 selects a base rule and binds, and then FD3.2 fires to solve for the unknown, then the effect is the same as FD2.1 selecting a derived rule and binding it. (Rules FD2.1, FD2.2, and FD3.2 are reproduced in Figure 4.5 for convenient reference).

---

FD2.1: Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

FD2.2: Given a set of attributes with known values then select a rule with those attributes as an input and an output and determine consistent entity bindings with one input unknown

FD3.2: If a bound computation rule has one unknown which is an input then solve for the unknown (isolate it on the left-hand side).

---

Figure 4.5. Strategic rules FD2.1, FD2.2, and FD3.2
4.2.7. Combination

Another way in which we form more complex domain rules is by combination. Combination forms a new rule from two or more existing rules. To form a combined rule, substitute the right-hand side (computation part) of a rule into one of the inputs or the output of another rule for a quantity that matches the left-hand side (output part) of the first rule. For example, when an airplane flies with the wind, in the standard computation rule

\[ \text{distance} = \text{net-rate-of-travel} \times \text{time} \]

the input net-rate-of-travel is replaced by the computation part of

\[ \text{net-rate-of-travel} = \text{wind-speed} + \text{airplane-speed} \]

to obtain the combination

\[ \text{distance} = (\text{wind-speed} + \text{airplane-speed}) \times \text{time}. \]

If the substitutions are made only for inputs to a base rule, then a new computation rule is formed that is more complex and computes the same output. Substituting for the output forms a new rule having a formula or equation instead of a computation part and an output. The process of combining rules has turned out to have more general applicability than solely to mathematical domains. Anderson (1986) employs composition (Lewis, 1978) in studying knowledge compilation in LISP programming. This composition is similar to the combination process described above. However, chapter 2 demonstrates the following distinction between combination and composition. Combination forms new rules that represent a hierarchical combination of component rules. Composition forms new rules that are a combination of a succession of component rules.

Composition combines rules that normally execute in sequence into a single rule. Therefore, composition is a linear combination because unique
predecessor and successor relations hold between the rules combined. Combination replaces components of rules by subcomponents. Therefore, combination is hierarchical because of these parent-child relationships, which may occur recursively.

To illustrate a combination rule, consider CR8 and CR3, reproduced in Figure 4.6. The output of CR3 matches all of the inputs and the output of CR8. Therefore, three combination operations result in an equation of the general form

\[ p_{ij}A_i = p_{2j}A_2 + p_{3j}A_3 \]

where \( p_{ij} \) and \( A_i \) are proportions of substance \( j \) in solution \( i \) and total amount of solution \( i \), respectively. Such an equation is exactly that which is applied by advanced solvers in solving "difficult" mixture problems.

---

**CR3:** The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.

Structure: Solution S with substance P
Constraints: none
Input values: amount(S), proportion(P)
Output value: amount(P)
Computation: proportion(P) * amount(S)

**CR8:** The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating substances.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively
Constraints: C1 not equal C2
Input values: amount(S2), amount(S3)
Output value: amount(S1)
Computation: amount(S2) + amount(S3)

---

Figure 4.6. CR3 and CR8.

The means-ends strategy is a more powerful strategy for forming combined rules from solution paths. The weak forward-directed strategy only
allows combinations to be formed when the problem solutions can be
generated in a forward manner by employing only one unknown. In the
following problem:

If you mix 10 gallons of a 30% acid solution with 20 gallons of a
15% acid solution, what is the percentage of acid in the mixture?
a sequence of weak forward-directed operations lead to the rule

\[ \frac{p_{11}A_1 + p_{21}A_2}{A_1 + A_2} \]

The weak forward-directed strategy does not have a rule-combining operation
like

ME3.2: Given two equations having a nonprimary goal variable in
common
then combine equations eliminating the nonprimary goal
variable.
of the means-ends strategy. This operation allows problems like

How much cream containing 35% butterfat should be added to
milk containing 4% butterfat to produce 155 qt. of half-and-half
containing 12% butterfat? (Groza, 1978, p. 192)
to be solved from base rules and the construction of the combined rule

\[ p_{11}A_1 + p_{21}A_2 = p_{31}A_3 \]
to take place.

**Pedagogical implications of combination rules**

Rules formed by combination are more easily derivable by applying the
means-ends strategy than by applying the weak forward-directed strategy.
This suggests that problems requiring rules formed by combination should not
be given as exercises to students who have progressed only to the weak
forward-directed level. This contradicts the pedagogy of Groza (1978), who
expects students to solve the problems presented in Figure 3.9 by employing
only one unknown and does not teach the use of two unknowns until later. Instead, Groza teaches students to solve these problems by exploiting a tabular form. In Table 4.1 we demonstrate how to solve the problem from the previous section by exploiting Groza's tabular formulation.

<table>
<thead>
<tr>
<th>Component</th>
<th>unit value</th>
<th>Amount (qt)</th>
<th>Value (qt)</th>
</tr>
</thead>
<tbody>
<tr>
<td>35% butterfat</td>
<td>.35</td>
<td>x</td>
<td>.35x</td>
</tr>
<tr>
<td>4% butterfat</td>
<td>.04</td>
<td>155-x</td>
<td>.04(155-x)</td>
</tr>
<tr>
<td>12% butterfat</td>
<td>.12</td>
<td>155</td>
<td>.12(155)</td>
</tr>
</tbody>
</table>

The entries in **bold** are those that are added to the table as part of the problem solution process.

Table 4.1. Groza's tabular solution of a complex mixture problem.

First the problem constants are written into the table, and then the empty positions in the table are filled. Finally, the equation necessary to solve the problem is formed from the last column

\[0.35x + 0.04(155-x) = 0.12(155).\]

This pedagogy appears to emphasize the teaching of addition in columns and multiplication in rows, which predominate in the solution, rather than emphasizing the teaching of reasoning about the legal combination of expressions. Explaining the tabular solution requires mention of the unknown amount of the 4% butterfat. Therefore, in reality there is another variable even though it can be expressed in terms of the first. The existence of the additional unknown makes the problem more suitable for techniques that employ two unknowns. Teaching the general problem solving strategy, means-ends, is consistent with the position that Bell (1980), Butts (1985), and others advocate: specific techniques (such as those of Groza) should not be taught for each problem type.
4.2.8. Deviation/correction

Deviation links connect correct rules to incorrect or deviant rules. For example, our data has several examples of students who attempt to sum proportions or percents from different solutions in the same manner as they would sum amounts. We represent the attempt to illegally combine proportions or percents in this way as an incorrect or deviant rule. Such a deviant rule is linked by a deviation/correct link to the correct rule for summing amounts. The deviation/correction link contains an explanation of the error and its genesis as a mistaken analogy that is exploited to tutor the student.

4.3. Strategy rule organization

Genetic links also connect the strategic rules into a genetic graph. Once again we connect the rules to make their interrelationships explicit for tutorial purposes. We hypothesize that these genetic interrelationships are the pedagogically proper relationships to make explicit.

4.3.1. Specialization/generalization

Instantiating a strategic rule is the specialization of the rule. For example, Figure 4.7 gives strategy rule FD2.1, a problem to which the rule may be applied, and computation rule CR1. When FD2.1 is given the attributes amount of the 10% alcohol solution and amount of pure alcohol, which have known values in the problem, it selects CR1, which has these attributes as its inputs, and it determines the bindings of 12 gallons of the 10% alcohol solution and 10 gallons of the pure alcohol to the inputs of CR1, then it is instantiated or
specialized. The strategic rule itself is the generalization of such instantiations.

**FD2.1:** Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

If you mix 12 gallons of a 10% alcohol solution with 10 gallons of pure alcohol, how many gallons will be in the total solution?

**CR1:** The amount of a solution is equal to the sum of the amounts of its parts.
- **Structure:** Composite solution C from solutions S1 and S2
- **Constraints:** S1 not equal S2
- **Input values:** amount(S1), amount(S2)
- **Output value:** amount(C)
- **Computation:** amount(S1) + amount(S2)

**CR3:** The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.
- **Structure:** Solution S with substance P
- **Constraints:** none
- **Input values:** amount(S), proportion(P)
- **Output value:** amount(P)
- **Computation:** proportion(P) * amount(S)

Figure 4.7. Examples for strategic rule specialization.

As was shown earlier for the computation rules, there are infinitely many specializations of strategic rules, because there are infinitely many possible problems in which to specialize the rule. Therefore, we represent the general rule and let the specializations be dynamically created by the rule interpreter when it unifies the variables of the general rule with the database which contains representations of the computation rules. This is another example of dynamic expansion of the genetic graph that Goldstein (1979) set as a goal.
4.3.2. **Analogies between specializations**

The solution of the problem in Figure 4.7 exhibits another use of the specialization of strategic rule FD2.1. When FD2.1 is given the attributes amount of the 10% alcohol solution and proportion of alcohol in the 10% alcohol solution, which have known values in the problem, it selects CR3, which has these attributes as its inputs, and it determines the bindings of 12 gallons of the 10% alcohol solution and proportion of .1 in the 10% alcohol solution for the inputs of CR3. This specialization of FD2.1 and the one in the previous section are analogous. Noting to the student that these two rule selections are analogous because they were selected by the same strategy rule (FD2.1) aids the understanding of the rule-selection process by highlighting that the same process is employed to select different computation rules to be applied in the problem-solving process.

4.3.3. **Refinement/simplification**

The refinement/simplification relationship plays an important role in the organization of the strategic rules. This relationship connects the rules into the four categories that were given in Chapter 3 and in Figure 3.3. No matter what strategic level a solver works at, the flow of processing in solving algebra word problems is between these categories. The grouping of rules into categories by the refinement/simplification relationship shows up in Figure 4.8 by the appearance of islands of related rules. In Figure 4.8 (as in Figure 4.2) we place the rule names in borders and link them with lines that are labelled either Refinement/Simplification or Analogy depending on which relationship holds between the rules. For example, Rule Selection is linked to each of FD2.1, FD2.2, ME2.1, ME2.2, EX2.1, and EX2.2 because each of these are
refinements of Rule Selection and Rule Selection is a simplification of each of them.
*These rules are not strategy rules in our model, but are connected by the refinements given to the named strategy rules of our model.

Figure 4.8. Graph of refinements and analogies for strategic rules.
Refinement

Simplification

ME3.2

Refinement

Simplification

Composition

Refine to
Single
unknown.

Refinement:
Unknown becomes
a goal.

Analogy

Categorization of the problem-solving process into rule-selection, solving-action, goal-establishment, and goal-reporting is not uniquely a result of defining refinement/simplification relationships for the strategic rules of
our model. For example, Bello's (1985) "procedure for solving word problems" given in Figure 4.9 gives a similar categorization with step 2 being related to goal-establishment, step 3 corresponding to rule-selection, step 4 corresponding to solving-action, and step 5 being related to goal-reporting.

1. Read the problem carefully and decide what is asked for (the unknown).
2. Select a variable to represent this unknown.
3. Translate the problem into the language of algebra.
4. Use the rules of algebra to solve for the unknown.
5. Verify the answer.

Figure 4.9. Procedure for solving word problems (Bello, 1985, p. 205).

An example of the refinement/simplification relationship for strategic rules is the relationship between rules ME1 and Goal Reporting given in Figure 4.10. ME1 is applied only when a subgoal is considered and not when a primary goal is considered. Therefore, ME1 applies in only a subset of the situations where Goal Reporting applies.

Goal Reporting: If a problem goal attribute has a value then note the solution and report the result.

ME1: If a problem subgoal attribute has a value then note the solution and report the result and propagate the result upward in the goal tree.

Figure 4.10. Strategic rules ME1 and Goal Reporting.

For example, in the problem
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

The primary goal, amount of pure alcohol, can be found from

\[ \text{amount pure alcohol} = \text{amount 70\% solution} - 10 \]

and the subgoal, amount of 70\% solution, can be found from

\[ \text{amount of 70\% solution} = \frac{8}{3}. \]

ME1 would apply to propagate the result for the subgoal, amount of 70\% solution, into the first expression. ME1 would not apply to the amount of pure alcohol when it is found because it is not a subgoal and there is nowhere to propagate the primary goal.

4.3.4. Analogies between refinements

Rules FD2.1 and FD2.2, repeated in Figure 4.11, are refinements of the simplified rule-selection rule. They differ only in the binding of a known quantity to an input as opposed to binding of a known quantity to an output. FD2.1 requires that two of the known quantities in the problems be legal inputs to the computation rule selected. FD2.2 differs only by requiring one of the known quantities be an output of the computation rule instead of an input. Writing the conditions out as a conjunction of predicates demonstrates how one can be mapped to the other by changing one condition. If we let \( Q \) be the set of known quantities with elements \( q_1 \) and \( q_2 \) and we let \( R \) be a computation rule, then the conditions of FD2.1 are written as

\[ q_1 \neq q_2 \land q_1 \in \text{inputs}(R) \land q_2 \in \text{inputs}(R). \]

By replacing the condition, \( q_1 \in \text{inputs}(R) \), with the condition, \( q_1 = \text{output}(R) \), we obtain the conditions of FD2.2

\[ q_1 \neq q_2 \land q_1 = \text{output}(R) \land q_2 \in \text{inputs}(R). \]
Therefore, FD2.1 and FD2.2 are analogous refinements of the same rule with a mapping between the conditions of applicability. Figure 4.8 diagrams this and related analogies by connecting rule names with lines marked Analogy.

FD2.1: Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

FD2.2: Given a set of attributes with known values then select a rule with those attributes as an input and an output and determine consistent entity bindings with one input unknown

Figure 4.11. Strategic rules FD2.1 and FD2.2

A more unusual form of analogy links rule FD2.1 with ME2.1 and rule FD2.2 with ME2.2. These rules are analogous because they refine to the same rule instead of being refinements of the same rule. Restricting FD2.1 to allow only goal entities to be the output produces the same rule as restricting ME2.1 to a single unknown (the goal). The restriction of FD2.1 to allow only goal entities to be the output changes the conjunction given above for the condition of FD2.1 to

\[ q_1 \neq q_2 \land q_1 \in \text{inputs}(R) \land q_2 \in \text{inputs}(R) \land \text{goal}(\text{output}(R)). \]

The condition part of strategic rule ME2.1 is

\[ \text{goal}(\text{output}(R)). \]

Restricting ME2.1 to a single unknown output means the inputs must be known, a condition which conjoined with the above condition gives the condition

\[ q_1 \neq q_2 \land q_1 \in \text{inputs}(R) \land q_2 \in \text{inputs}(R) \land \text{goal}(\text{output}(R)). \]

This derivation showing that it is possible to refine the conditions of FD2.1 and ME2.1 to the same condition demonstrates that FD2.1 and ME2.1 are analogous in the manner introduced above.
4.3.5. Composition/component

An example of composition is EX3 which has as its components modified versions of ME3.1 and ME3.2 (These rules are repeated in Figure 4.12 for your convenience). EX3 solves two equations in two unknowns whereas ME3.2 eliminates an unknown from two equations and ME3.1 solves for the remaining unknown.

---

ME3.1: Given a goal attribute
and an equation (a bound computation rule) with the goal variable
then solve the equation for that variable (isolate it on the left).

ME3.2: Given two equations having a nonprimary goal variable in common
then combine equations eliminating the nonprimary goal variable.

EX3: Given two equations with the same two unknowns
then solve for the two unknowns.

---

Figure 4.12. Strategic rules ME3.1, ME3.2, and EX3.

For example, given the equations

\[ 10 + x = y \]

and

\[ 2 + x = .7y \]

from

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

ME3.2 would reduce them to

\[ 2 + x = .7(10 + x) \]

and ME3.1 would produce

\[ x = \frac{5}{3} = 16.6. \]
This example shows that EX3 is not a pure composition of ME3.1 and ME3.2. This presently is not a problem because the links currently hold fixed textual information that explains the relationship whether pure or not. Further development and refinement of all the rules would have as a goal the establishment of their exact relationships. These relationships could have dynamically generated explanations from a formal definition of the relationship, whether it be composition or some other relationship. However, it is not clear that developing and implementing all refinements of applicable rules should be a goal. After all, learning presently takes place with the highly simplified rules applied by teachers and textbooks. For example, consider the simplified rules of Bello (1985) presented above.

4.3.6. **Sophistication**

Strategic levels of expertise are joined by explicit sophistication links to direct a path to increased ability. For example, FD2.1 is joined, with a sophistication link, directly to ME2.1. This link, in conjunction with the analogy link between the two rules that was discussed above, can be exploited to explain

* that ME2.1 is part of a more sophisticated strategy (exploiting the sophistication link), and

* how ME2.1 is related to the rule FD2.1 that the student already applies (exploiting the analogy link).

4.4. **Factors in the genetic graph demonstrating difficulty**

In the previous chapter we showed that problem difficulty corresponds to the level of strategic expertise necessary to solve the problem. We analyzed a set of problems (believed by Groza (1978) to be increasing in difficulty) with respect to how well application of the weak forward-directed, means-ends, and
expert strategies handles solving them. The result of the analysis was a demonstration that the progression of strategy levels matched the progression of difficulty in the problems. Characteristics of domain rules determine problem difficulty, as well. In the following sections we illustrate factors, such as rule derivation and rule combination, which affect rule difficulty.

4.4.1. Derived rules

We informally tested the hypothesis that problems requiring derived rules for solution are more difficult to solve than those requiring only base rules for direct solution. We tested the hypothesis first on students before instruction by giving a test with the problems in Figure 4.13 to each of two sections of an intermediate algebra class on their first day of class at the University of Portland.

1. What is the volume of acid in 40cc of a 5% acid solution?
2. What is the percentage of alcohol in a 40-quart alcohol-water solution that has 15 quarts of alcohol?
3. What is the weight of copper in 60 pounds of copper ore that is 15% copper?
4. What is the weight of the ore if you have 12 pounds of copper in an ore that is 4% copper?
5. What is the percentage of copper in 120 pounds of copper ore that has 6 pounds of copper in it?
6. What is the volume of alcohol-water solution you have if you have 8 quarts of alcohol in a solution that is 5% alcohol?

Figure 4.13. Simple mixture problems from intermediate algebra pretest.

Our hypothesis would suggest that problems 1 and 3 would be solved correctly by the most students because they can be solved by directly applying
CR3 (which we consider to be the base rule). Problem 1 applies CR3 to set up the computation \(0.5 \times 40\), and problem 3 applies CR3 to set up the computation \(0.15 \times 60\). The other problems can be solved by directly applying CR4 or CR5.

Therefore, we hypothesize that they are more difficult because rules CR4 and CR5 are derived from rule CR3. CR4 is applied to set up problem 4 as \(\frac{12}{0.04}\), and problem 6 as \(\frac{8}{0.05}\). CR5 is applied to set up problem 2 as \(\frac{15}{40}\), and problem 5 as \(\frac{6}{120}\).

The problems were ordered so that the same rule would not apply directly twice in succession, and so that the same sequence of rules would not apply twice. We employed a X² test on the hypotheses

\[
H_0: \text{No difference in difficulty between problems requiring CR3 and problems requiring CR4}
\]

\[
H_a: \text{Problems requiring CR4 are more difficult than problems requiring CR3}
\]

The conclusion of this test was to reject \(H_0\) with \(\alpha < 0.0025\).

In the remaining problems on the test, two or more applications of computation rules were required. These problems proved too difficult for the students in these classes. The students were told that the test was an attempt to determine their level of ability, and that the results would not be employed in determining their grade for the course. The results from the 18 students in the two classes are given in Table 4.2.

<table>
<thead>
<tr>
<th>Problem number</th>
<th>rule that solves</th>
<th>number correct</th>
<th>number attempted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CR3</td>
<td>7</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>CR5</td>
<td>8</td>
<td>17</td>
</tr>
<tr>
<td>3</td>
<td>CR3</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>4</td>
<td>CR4</td>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>5</td>
<td>CR5</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td>6</td>
<td>CR4</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 4.2. Intermediate algebra pretest results.
Our hypothesis was substantiated clearly in the case of CR4 but not in the case of CR5 (for which there were only two fewer correct solutions than for CR3).

We explain this result by noting that computing the percent of a solution is a common operation. Therefore, CR5 should be given the status of a base rule as a "familiar" computation.

We gave the first of the two midterm tests to be given during the semester after several weeks of instruction in intermediate algebra while employing the text by Phillips et al. (1983). The text emphasizes word problems or "applications" as a motivation for learning algebra. Thus the students had been exposed to a continuing stream of word problems in lecture, in homework, and on quizzes. We argue that if problems solvable by derived rules continue after instruction to be solved correctly less often than those solvable by base rules, then we have stronger confirmation of our hypothesis, because these problems continue to be more difficult for the students than those solvable by base rules thereby demonstrating their inherent difficulty.

For the first midterm, the first section of the class was given the problems shown in Figure 4.14. Problem 21 is solvable by applying CR3 to set up \( .25 \times 2 \times 100 \), problem 22 is solvable by applying CR5 to set up \( \frac{20}{80} \), and problem 23 is solvable by applying CR4 to set up \( \frac{10}{.4} \). The results from the 1924 students in this section are given in Table 4.3.

---

24 The reason there are more students is that a pretest had been given in higher level classes and several "better" students were dropped back to the intermediate algebra class.
21. How much acid is in 100 gallons of a 25% acid solution?

22. If 20 gallons of an 80-gallon acid solution are acid, what is the percentage of the acid?

23. If 10 gallons of a 40% acid solution are acid, how much solution is there?

Figure 4.14. Mixture problems from section 1 midterm.

<table>
<thead>
<tr>
<th>Problem number</th>
<th>rule that number solves</th>
<th>number correct</th>
<th>number attempted</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>CR3</td>
<td>10</td>
<td>15</td>
</tr>
<tr>
<td>22</td>
<td>CR5</td>
<td>10</td>
<td>16</td>
</tr>
<tr>
<td>23</td>
<td>CR4</td>
<td>6</td>
<td>14</td>
</tr>
</tbody>
</table>

Table 4.3. Results from first midterm given to first section.

After the pretest we expected problem 22 to be about the same difficulty as problem 21. As can be seen in the table, the same number of students got these two questions right. Our other prediction that problem 23 would be more difficult than problem 21 was tested with a $X^2$ test as above. The test resulted in $\alpha = .16$, which is not significant enough to reject the possibility of obtaining the above results by chance. However, given that these students had been instructed on and studied these types of problems, we argue that there is reason to believe that problems requiring CR4 are more difficult.

We employed the first midterm in the second section to attempt to further confirm the hypothesis with problems from a different domain than mixture problems. On this midterm we gave the distance-rate-time problems given in Figure 4.15. Problem 21 is directly solvable by applying

$$\text{distance} = \text{rate} \times \text{time}$$
which we consider to be the base rule for this domain. Problem 22 is directly solvable by applying

\[
\text{rate} = \frac{\text{distance}}{\text{time}},
\]

and problem 23 is directly solvable by applying

\[
\text{time} = \frac{\text{distance}}{\text{rate}}.
\]

Therefore, according to our hypothesis, problems 22 and 23 should be more difficult to solve than problem 21. However, on each problem all 11 students who answered the problems solved them correctly. The one remaining student did not attempt any of the problems.

21. If you drive at 60 mph for \(2\frac{1}{2}\) hours, how far have you driven?

22. Amtrak takes 24 hours to travel the 1200 miles between Portland and Wolf Point, Mont. How fast does the Amtrak train travel?

23. San Francisco is 638 miles from Portland. I drove at 58 miles per hour from San Francisco to Portland. How long did it take me?

Figure 4.15. Distance-rate-time problems from section 2 midterm.

We offer two interpretations of this result. If familiarity determines what is a base rule, then this domain is familiar enough that all the forms of the rule are base rules. These students were simply too good, especially after instruction in the topic, for the difference to be shown is our second interpretation. Noting that the difference did show up in the other section taught during the same time period by the same instructor makes this interpretation questionable. However, the conversion of percent to proportion complicated the other problems and made them slightly more difficult.
We stand by our conclusion that problems differ in difficulty corresponding to whether they can be solved by base or derived rules. Indeed all but one of the students who applied a formula, instead of direct application of a rule, applied the formula

\[ d = r \times t. \]

The other formula applied was

\[ r = \frac{d}{t} \]

to solve for time, after which the formula was manipulated to isolate the one unknown, time, on the left-hand side. Therefore, problems requiring

\[ r = \frac{d}{t} \text{ and } t = \frac{d}{r} \]

are more difficult because students set them up with

\[ d = r \times t \]

and employ an extra solution step.

4.4.2. Combined rules

Problems whose solutions require more than one computation rule to effect a solution appear to be more difficult than those which do not. The tests we gave the first day of the intermediate algebra class graphically illustrated this. In addition to the six problems discussed previously, the tests contained seven problems requiring combinations of rules. Students attempted only 37% of these seven problems, whereas students attempted 79% of the first six problems. Of the attempts on the last seven problems, 15% reached the correct answer, whereas 38% of the attempts on the first six problems reached the correct answer. If we include the problems that were not attempted, then the
success rate for the first six problems falls only to 30% whereas the success rate for the final seven problems falls more drastically to 6%.

4.4.3. Other difficulty factors

Other factors determine rule difficulty besides derivation and combination. In addition to demonstrating the derivation and combination links, Figure 4.16 depicts rule difficulty by dividing an expanded rule set into classes of difficulty with a characterization of the difficulty of each class. The figure shows how partial specialization affects rule difficulty by decreasing the number of inputs to bind. For example, the specialization

\[ a = 0 \times A \]

of CR3 does not require that any attempt be made to ascertain (or bind) the value of \( A \), as the value of \( a \) will be 0 whether or not the value of \( A \) is known.

A related manner in which specialization affects problem difficulty is the elimination of the need to carry out an operation. If the value of a quantity is the additive identity zero, and it is to be employed in a computation that employs addition or in a computation in which it is subtracted from another quantity, then the operation does not have to be performed. This is likewise true when the multiplicative identity one is employed in conjunction with multiplication or division. For example, when a problem such as

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

refers to a pure solution, the amount of the pure substance, \( a \), in the pure solution is equal to the amount of the pure solution, \( A \). This can be seen in the specialized rule for CR11

\[ a = A - 0. \]
This indicates that the amount of the pure solution, \( A \), can be taken directly as the amount of the pure substance, \( a \), without taking anything away because of another substance, thereby eliminating the need to carry out an operation.

Figure 4.16 shows the spectrum of difficulty of computation rules. In the figure the easiest computations are placed on the left, and the hardest computations are placed on the right. The easiest computations are those for which the value of one of the inputs determines the constant output without any computation needing to take place (e.g. in the specialization of CR3 discussed above). Next in the spectrum come the computations in which the value of one of the inputs determines that the problem solver must output the other input. Because one of the values employed in CR6 and CR7 is the constant full proportion 1.0, these rules are classified as falling next in the spectrum. In the next segment of the spectrum, we place the base computation rules to the left of the computation rules that we derived from the base computation rules. This placement is done to designate the increase in difficulty that we attribute to derived rules. The succeeding segments of the spectrum give examples of combined rules. Each segment combines more rules on the progression to the most difficult rules on the right.
Figure 4.16. Specialization, derivation, and combination links for wet-mixture domain rules depicting levels of difficulty.
4.5. Providing a syllabus

The genetic graph shows the evolution of strategic rules. The categories of rules evolve by refinement from simple rules for solving word problems such as rules 3 and 4 of Bello (1985) in Figure 4.9. The genetic graph in Figure 4.2 shows the evolution of computation rules from the common concepts of addition and multiplication. Both the graph for the strategic rules and the graph for the computation rules have an implicit difficulty gradient embedded in them. A student's movement along these difficulty gradients leads to achievement of greater proficiency. Therefore, the genetic graph provides a syllabus that indicates the relationship of the chunks of knowledge and provides direction to increase both domain knowledge and strategic ability.

Beginning students enter the graph for the strategic rules at the weak forward-directed level. We assume these students already have some knowledge of applying FD1, FD2.1, and FD3.1 from solving arithmetic word problems. Students then achieve proficiency by traversing the graph level by level.

Instruction of the base rules causes students to enter the computation rules at that level. During traversal of the graph of computation rules, the students experience movement in the direction of easier rules because of special cases, and they experience movement towards the more sophisticated derived and combined rules in concert with increased strategic competence.
4.6. Providing a student model\textsuperscript{25}

Employing the problem-solving model described in chapter 3, one can form an *overlay* (Goldstein and Carr, 1977) of the knowledge in the model, that is a model constructed from a portion of the problem-solving model. Therefore, the student model is a subset of the entire model which encompasses novice (including erroneous) through expert behavior. An enhancement to the model provided by the genetic links is the delineation of the frontier of a student's knowledge. Delineating the frontier of a student's knowledge provides a model of what knowledge a student can assimilate next with the most ease. We base this claim on the assumption that a student will assimilate a closely related concept with more ease than he will assimilate a more distantly related concept. Our measure of the closeness of one concept to another is the number of links in the genetic graph separating the rules representing the concepts.

4.7. Providing tutorial capability

The genetic graph facilitates tutoring in two ways. First, it makes explicit the relationship of a new concept to known concepts. This allows explanation to be made based on the relationship between the new concept and the known concept. Because often there is more than one such relationship available, alternative explanations are available. Second, the graph facilitates tutoring by suggesting, from the model of the student's knowledge, that the next topic to be tutored is on the frontier of the graph.

\textsuperscript{25}Chapter 6 discusses student modeling.
Goldstein (1979) points out that intelligent tutoring systems can respond to a larger number of situations than script-based computer-assisted-instruction. This is because the response need not be preprogrammed, but arrived at dynamically by the embedded expert when it processes the current state. The genetic graph allows multiple responses in these situations. For example, if the system needs to explain CR2 to a student, it has several choices on which to base its explanation. Its choices are to explain CR2 in terms of CR1, from which it is derived; in terms of CR11 or CR9, which are analogous; in terms of the simplification, which is not specific to amounts of solutions; in terms of its specialization in the problem at hand; or in terms of an analogous prototypical specialization. The genetic links that are represented explicitly have explanatory text stored with the link as shown in Figure 4.17. The links that are represented implicitly by rules standing for the process of establishing the link could easily be supplemented with "canned" explanations. Tutorial strategy determines the choice of which explanation to employ.
Figure 4.17. Genetic links with explanatory text.

Script-based CAI allows the establishment of a meaningful but rigid order for introduction of subject matter. Intelligent tutoring systems are built to respond to the situation. Therefore, intelligent tutoring systems eliminate the rigidity, but lose the topic ordering. In an expert based tutor, the genetic graph retains the flexibility of an intelligent tutor and suggests meaningful topics (Goldstein, 1979).

In Section 3.2, we presented four approaches to tutoring a student based on his knowledge and ability. Of these, three attempt to increase knowledge or ability and, therefore, would be aided by topic suggestions from the genetic graph. Two of these three call for tutoring with the goal of expanding the student's computation rule knowledge. Expanding the student's computation rule knowledge is possible within a single domain or by extending to other domains. Extending to other domains demonstrates the generality of the
strategic rules. The third approach calls for increasing the student’s strategic competence. For example, consider the student protocol that was run in Figure 3.12. In this protocol the student solved

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to obtain a 70% solution?

by finding the amount of the 70% solution was 26.6, employing the 26.6 to find the amount of alcohol in the 70% solution was 18.6, and then subtracting the two gallons of alcohol in the 20% solution to get 16.6 gallons of pure alcohol. Applying CR2 would have allowed the student to subtract the 10 gallons of 20% solution from the 26.6 to immediately arrive at the answer. Because this is the only problem this student worked on, there is no evidence that the student knows CR1 and CR2. This lack of evidence exists even though CR1 and CR2 are near the entry to the genetic graph of computation rules. Therefore, CR1 and CR2 are certainly on the student’s genetic graph frontier and the system could choose to tutor the application of CR2 at this point to improve the student’s solution. Also, Figure 3.12 showed that al-Khorezmi determined that the student had the ability to perform at the means-ends level. Therefore, another alternative is for the system to suggest proceeding on this frontier showing how subgoaling allows a solution in terms of the alcohol, instead of applying the weak forward-directed strategy and solving by employing quantities of water as the student did.

4.7.1. Providing a passive help system

Earlier we discussed the active help that is available in the form of leading question hints. In addition we provide a passive help facility. This passive facility allows students to explore knowledge on their own. Because the genetic graph organizes the domain by explicitly representing the
relationships between information, we allow students to traverse the knowledge base by following the genetic links. Therefore, we provide a help system that gives the students a choice of examining rules they have or have not applied. After a student selects a rule, al-Khorezmi allows him to select what kind of related rule he would like to see. Then al-Khorezmi chooses a link of that type from the selected rule and presents a related rule. In the next sections we show student employment of this facility and demonstrate portions of the facility not exploited by students in the experimental use.

Use of passive help by students during online session

In our experimental test of al-Khorezmi, two students exploited this help system to obtain help successfully. The students that exploited the help facility seemed most interested in obtaining immediate help. Because they were employing al-Khorezmi only for the duration of a 42-minute class period, their interest in obtaining immediate help was heightened. Therefore, they chose to see rules applicable to the current problem-solving situation followed by specializations of these rules. This allowed them to see how al-Khorezmi would proceed in solving the problem. One other student asked for help early enough in the session to have time to receive beneficial help. Al-Khorezmi presented him with the overly general rule discussed in 3.1.4 followed by many specializations of that rule which were not useful to the student. All other students who needed help encountered their difficulties at the end of the period. These students ran out of time before receiving sufficient help.

In Figure 4.18 we present the interaction by student DH who experienced difficulty using al-Khorezmi and solving the problems from the very beginning. After the experimenter explained how to run al-Khorezmi,
DH still asked for help in running the program and had trouble noting a constant to al-Khorezmi. The help offered demonstrates that al-Khorezmi needs a refinement of rule-selection rules FD2.1 and ME2.1. This is because it is of no value to present CR7 when the goal value is directly computable (as is done at point (1) in Figure 4.18). The refined rule, being more specific and therefore given a higher rating in the conflict set of the production system monitor, would fire and avoid the presentation of CR7. Before the experimental use of al-Khorezmi we had no evidence that the refined rule would be necessary. FD2.1 or ME2.1, both being simplifications of the needed rule, performed essentially the same function in the place of such a rule.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Which option would you like to have explained? 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10 ? 1

At various stages in solving a problem there are quantities you want to calculate or solve for. This option should be used to note to the system which quantities you are trying to obtain. You will be prompted for Quantity? to which you should answer with the letter from the menu of quantities.

Option? 1 Quantity? b
OPERATION VERIFIED TO BE CORRECT

Option? 2 Quantity? f Value? 10
That constant belongs to quantity amount-20%-solution.

Option? 2 Quantity? c Value? .2
OPERATION VERIFIED TO BE CORRECT

Option? 9

Would you like help on :

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu
   ? 1

A rule applicable to your request is FD2.1: if THERE IS A RULE WITH KNOWN VALUES FOR ALL ITS INPUTS then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS.

---

Figure 4.18. Student DH's interaction with al-Khorezmi.
Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) More rules that satisfy your last request (if any)
6) a return to the previous menu

1

**[2]** THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE PROPORTION OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0 is a rule that has as input the known entities 1.0 & proportion-alcohol-20%-sol.

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2

THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION is a rule that has as input the known entities proportion-alcohol-20%-sol & amount-20%-solution.

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 1

amount-alcohol-20%-sol( ) = proportion-alcohol-20%-sol( ) * amount-20%-solution( )

Would you like to see values with that equation (y or n)? y

amount-alcohol-20%-sol(Uknown) = proportion-alcohol-20%-sol(0.2) * amount-20%-solution(10)

Would you like to 1) return to the previous menu, 2) see another example of the strategy? 1

Would you like help on:

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu

? 5

Option? 4 Quantity? b Value? 2
OPERATION VERIFIED TO BE CORRECT

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason ? 1

Would you like to 1) stop now 2) load a new problem ? 1

Figure 4.18. (continued)
Demonstration of help facilities not exploited by student subjects

In Figure 4.19 we present a specific traversal of some of the other links in the genetic graph to demonstrate that capability. At point \{1\} the user begins the exploration of the rule set by requesting a presentation of new computation rules. Al-Khorezmi responds with CR10. The user continues his exploration at point \{2\} by asking for an analogous rule. The response presents CR1 and a "canned" explanation of its relationship to CR10. Next the user asks for a simplification at \{3\} and gets the statement of a more general rule. Asking for a more specific rule at \{4\} causes al-Khorezmi to follow the refinement link back to CR10. At \{5\} the user asks for the origin of CR10 and finds that it is a base rule and so tries to see what is derived from it at \{6\}. In response, the related rule CR11 is printed with a "canned" explanation of the relationship requested. At this point the help is terminated at the user's request.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

<table>
<thead>
<tr>
<th></th>
<th>a) amount-30%-solution</th>
<th>b) amount</th>
<th>c) proportion</th>
<th>d) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>acid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>e) amount</th>
<th>f) proportion</th>
<th>g) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>h) amount-15%-solution</th>
<th>i) amount</th>
<th>j) proportion</th>
<th>k) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>acid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>l) amount</th>
<th>m) proportion</th>
<th>n) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>o) amount-mixture</th>
<th>p) amount</th>
<th>q) proportion</th>
<th>r) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>acid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>water</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>s) amount</th>
<th>t) proportion</th>
<th>u) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

v) full proportion constant 1.0  w) full percent constant 100.0  y) other value

1) note goal  5) give expression  9) help-algebra
2) note constant  6) solve  10) exit-problem
3) specify a variable  7) help-hints
4) give calculated result  8) help-menu-options

Option? 2 Quantity? a Value? 10
OPERATION VERIFIED TO BE CORRECT

Option? 2 Quantity? h Value? 20
OPERATION VERIFIED TO BE CORRECT

Option? 9

Would you like help on:

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu

A rule applicable to your request is cr10: WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF THE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SUBSTANCES

Figure 4.19. Demonstrating traversal of links other than specialization.
Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

A rule applicable to your request is cr1: WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF THE COMPOSITE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE CONSTITUENT SOLUTIONS. This rule is analogous to the previous one, it uses amounts of solutions in a composite instead of amounts of substances in a solution.

Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

A rule applicable to your request is simplified-mixture-sums: WHEN TWO THINGS ARE COMBINED, THEIR AMOUNTS ARE SUMMED. REFINEMENTS OF THIS RULE THAT SPECIFY WHICH THINGS ARE TO BE SUMMED CAN BE APPLIED TO MIXTURE PROBLEMS. This rule is a simplification of the previous one, it does not distinguish what amounts are to be added.

Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

A rule applicable to your request is cr10: WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF THE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SUBSTANCES. This rule is a refinement of the previous one, it specifies that the amounts of the substances in a solution are added.
Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

There are no origination relationships between cr10 and other rules for viewing.

Would you like to
1) have a chance to follow up on the same rule
2) go on to a new rule
3) return to previous menu?
1

A rule applicable to your request is cr10: WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF THE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SUBSTANCES.

Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) A rule derived from this rule (if any)
6) A rule this is derived from (if any)
7) More rules that satisfy your last request (if any)
8) a return to the previous menu

A rule applicable to your request is cr11: WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF ONE OF THE SUBSTANCES IS FOUND BY SUBTRACTING THE AMOUNT OF THE OTHER SUBSTANCE FROM THE AMOUNT OF THE SOLUTION. This rule is derived from the previous one, it finds one of the parts by subtraction instead of the whole by addition.

Figure 4.19. (continued).
4.7.2. **Towards a more general tutorial**

The present tutoring ability of al-Khorezmi is limited to two options: suggesting a proper way to proceed when the student becomes stuck and allowing the student to explore the knowledge base on his own. In Appendix D, we present the design of an extended interface that exploits the genetic graph to gain tutorial leverage.

In our student-teacher protocols, we have observed examples of tutoring that we can exploit to increase tutorial leverage of al-Khorezmi. When the students become stuck, our protocols show tutors making up a new problem that they think the students can solve to demonstrate the application of a needed computation rule. The tutors' goal was to get the students to apply the computation rule both in the new problem and then in the problem at hand. In this case, the tutors are taking advantage of the analogy relationship...
between two specializations of the same rule. Another way to exploit these analogies in a computer tutor is to maintain a database of a student's problem solutions that can be retrieved in situations requiring another application of a previously applied rule. Then, when the student gets stuck the system can retrieve the application of the desired computation rule from a previous problem solution and present it to the student to remind him of how to perform the step that is necessary in the current problem. Another tactic employed by tutors is to make up examples that will cause a student to see the fallacy of his methods when he makes a mistake. These are examples of tutorial tactics involving issues such as when to regress or progress and how to handle such regression or progression. The organization of knowledge "inherent" in the genetic graph serves to guide the selection of material whether it is to be exploited for progressing or regressing.26

4.8. Relation to Wusor

Goldstein (1979) views the employment of the genetic links as a way of extending the ability of a system to model a student. This extension goes beyond previous systems which employed a subset of expert skills to model a student. The extensions include modeling student abilities with specializations, deviations, and simplifications of expert rules.

We complement this notion by deriving our student model from protocol data as opposed to deriving a model from expert behavior. Naturally, the model derived from protocols may include specializations, deviations, and simplifications of expert rules. However, these relationships do not completely cover the differences between expert and novice models. We support this

26This was clearly demonstrated in Section 4.7, which showed the multiple ways CR2 could be presented.
statement by noting that the expert strategic rules of our model can all be formed from compositions of the means-ends rules in our model. These expert rules must be separated into components before exploiting refinement/simplification and analogy to form lower level student rules. Therefore, in our model it is not simplifications of expert behavior that students use but components of expert behavior and analogous weaker methods. For example, in Figure 4.20 we give EX2.2 which has as two of its components ME4 and a rule-selection rule similar to ME2.1 or ME2.2 (also given in Figure 4.20). Because we have observed students applying ME2.1 and ME2.2 instead of EX2.2 we conclude they are employing component skills. Likewise, students apply FD2.1 and FD2.2 which are analogous rules of a weaker method. A similar phenomenon appears in the computation rules. The expert applies rules that are combinations of the rules that students apply.

EX2.2: Given an unknown goal attribute
and a bound computation rule employing it and another unknown
then select another computation rule with those unknowns
and determine consistent entity bindings.

ME2.1: Given a goal attribute
then select a rule computing its value as output
and determine consistent bindings.

ME2.2: Given a goal attribute
then select a rule having it as input
and determine consistent bindings.

Figure 4.20. Strategic rules EX2.2, ME2.1, and ME2.2.

Because of the simplicity of a domain like the game of Wumpus, the construction of a genetic graph for that domain highlights different rule relationships from those highlighted by the construction of a genetic graph for a problem-solving model for algebra word problems. The game allows only
moving one cave at a time followed by presenting information for determining the next move "locally" after each move. Moving only one cave at a time is like a student solving arithmetic word problems by applying strategy rules FD1, FD2.1, and FD3.1 to solve one quantity at a time. It is not like solving algebra word problems where the "global" statement of the problem determines all moves. Instead of making one move at a time, experts progress to the point of making multistep jumps. This characterization shows why learners' rule sets cannot be constructed simply, as Goldstein proposes, "from specializations, deviations, and simplifications of expert's rules" (1979, p. 59).

Another factor, that causes a difference in determining the important relationships of the genetic graph, is the fuzzy notion of what is a mistake in a game versus the clear-cut distinction between correct and incorrect operations in mathematics. In mathematics, simplifications give erroneous rules more often than they do in the game. When a simplified rule is applied in the game, play may be continued without correction, as long as an immediate loss does not occur. This is not so when a simplified rule is applied in mathematics, yielding an incorrect operation.
5. Corrective Feedback

5.1. Introduction

In this chapter we describe our analysis, modeling, and correction of erroneous behavior. We model the origin of bugs with the genetic relationships of the genetic graph and show how the simplification relationship explains all bugs found in the analysis of our data. Furthermore, we relate our bug explanations to the work of Matz (1982) and Brown and VanLehn (1980) and show that our problem-solving model could be improved by incorporating self-monitoring skills to fully model human problem-solving skills.

To show the extent of our coverage of errors we relate our work to Goldstein who reported that they represented "Correct application of incorrect rules" (1979, p. 58). We show how we extend the representation to represent incorrect application of correct rules and incorrect application of incorrect rules. Distinguishing correct rules applied incorrectly from deviant (incorrect) rules allows a more economical representation than encoding multiple deviations (of the correct rules) to be applied correctly. The distinction also characterizes significantly different error types. The correct rules applied incorrectly characterize the incorrect binding of an operand or operator in a rule, whereas the incorrect rules applied correctly characterize the incorrect binding of all operands in an analogous rule. Furthermore, these two error types combine in only one way to form incorrect rules applied incorrectly that explain otherwise unexplained student errors found in our data.
5.2. Study of errors in recorded problem solving

Error detection and correction is important to tutoring a procedural skill. To correct errors, an intelligent tutoring system must have knowledge of typical errors and their genesis. We approached supplying the system with this knowledge in two ways. First, we studied student work in student-tutor protocols and on tests given to the intermediate algebra class at the University of Portland. From this study we encoded the errors we found and the corrections of those errors into the program by employing either a static description or a dynamic expansion process.

Collecting data on errors students make while working on algebra word problems is a laborious task. In the future we plan to have al-Khorezmi permanently record all errors students make and particularly any errors it cannot explain adequately during an interaction. Al-Khorezmi cannot explain an error adequately if no component of its error model matches the error (in which case it has no explanation) or if its error model proposes more than one explanation for the error. Presently, we treat recorded interactions between students and al-Khorezmi as data and search for unexplained errors to analyze. Interestingly, our experimental use of al-Khorezmi at Lutheran High School did not uncover any additional error types, since al-Khorezmi explained all errors made by these students.

In the following subsections we discuss our studies of student errors and present our ideas on the genesis of these errors. We then compare these ideas to other research on the genesis of errors.

---

27 Chapter 2 emphasizes this importance by making corrective feedback the first of two primary pedagogical principles.

28 Having the problem-solving interaction recorded by the system is a great labor saver by itself.
5.2.1. Errors in simple wet-mixture proportion problems

The tests we collected from the intermediate algebra class had fifty-five errors on attempts to solve problems requiring direct application of computation rules CR3, CR4, and CR5. Of these fifty-five erroneous attempts, forty-five were attempts by students to apply either one of CR3, CR4, or CR5; one was an attempt to apply CR7; and the other nine were attempts to apply a combination of rules made up of CR3, CR4, and CR5 (wet-mixture problem computation rules are given in Figure 5.1).

First, we present the errors observed in the forty-five attempts to apply CR3, CR4, and CR5. We classify the solution attempts into two categories:

1) attempts to apply the correct rule, and

2) attempts to employ the operator of a rule connected to the correct rule by a derivation/origination link.

These are further broken down into categories based on the number of incorrect usages of quantities. Table 5.1 gives the tabulation of the number of observed occurrences within these categories for students attempting to apply the correct rule.

---

29 Actually, there were additional operations--converting percent to proportion and converting proportion to percent--but only one of the fifty-five erroneous solution attempts explicitly showed an attempt to do these operations which are normally done as spontaneous computation. Therefore we omit these operations.
CR1: The amount of a solution is equal to the sum of the amounts of its parts.

CR2: The amount of one part of a solution is equal to the amount of the solution minus the amount of the other part.

CR3: The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.

CR4: The amount of a solution can be found by dividing the amount of the substance in the solution by the proportion of the substance in the solution.

CR5: The proportion of a substance in a solution can be found by dividing the amount of the substance in the solution by the amount of the solution.

CR6: The sum of the proportions of the substances in a solution is 1.0.

CR7: The proportion of a substance in a solution is 1.0 minus the proportion of the other substance in the solution.

CR8: The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating solutions.

CR9: The amount of a substance in a solution is the amount of that substance in the composite solution minus the amount of that substance in the other solution in the mixture.

CR10: The amount of a solution is the sum of the amounts of the two participating substances.

CR11: When two substances are mixed to form a solution, the amount of one of the substances is found by subtracting the amount of the other substance from the amount of the solution.

Figure 5.1. Computation rules for wet-mixture problems.
No incorrect quantities employed
inputs bound in reverse order (noncommutative operation) 4

One incorrect quantity employed
bound to an input, other input correct 4
bound to an input, inputs reversed 2
bound to the output, inputs correct 0
bound to the output, inputs reversed 6

Two incorrect quantities employed 0

Table 5.1. Categorization of errors in applying the correct rule.

To explain these errors in terms of genetic relationships, we note that a simplified form of each rule will not specify the exact binding of the various quantities. Employing a simplification allows the student to form his bindings from incomplete knowledge. For example, the first problem given in Figure 5.2 can be solved by applying rule CR4 given in Figure 5.2. One of the students attempted to solve this problem with

\[ A = \frac{10}{40}. \]

Removing the condition in rule CR4, that one of the inputs be bound to proportion(P), opens up the possibility of the student binding that input to another quantity such as the quantity, percentage(P), that was bound in this example. The other fifteen erroneous solution attempts by students are similar, so much so, that the hypothesis--students employ simplifications of correct rules--is a plausible explanation for all sixteen errors categorized in

---

This characterization of errors captures how a rule has been misused, but not the relationship between the correct and incorrect quantities. The current characterization of errors can be extended in the future to consider the relationship between these quantities.
the table above. In fact, we can constrain the simplification even further--to allowing only one of the quantities of the problem to be incorrectly chosen--and still explain all the errors in this situation.

Problem 1:
If 10 gallons of a 40% acid solution are acid, how much solution is there?

Problem 2:
How much alcohol is there in 200 gallons of a 45% alcohol solution?

CR3: The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.

| Structure: | Solution S with substance P |
| Constraints: | none |
| Input values: | amount(S), proportion(P) |
| Output value: | amount(P) |
| Computation: | proportion(P) * amount(S) |

CR4: The amount of a solution can be found by dividing the amount of the substance in the solution by the proportion of the substance in the solution.

| Structure: | Solution S with substance P |
| Constraints: | none |
| Input values: | amount(P), proportion(P) |
| Output value: | amount(S) |
| Computation: | amount(P) + proportion(P) |

Figure 5.2. Two mixture problems with applicable computation rules.

In solutions that employed as their operation an operator from a rule connected by a derivation/origination link, the order in which the inputs are bound is not a significant distinguishing characteristic because one or the other of the operations is commutative. For example, one of the students attempted to solve the second problem in Figure 5.2 with

\[ a = \frac{200}{.45} \]
Because the problem would normally be solved with CR3 which employs the commutative operation multiplication, the appearance of 200 before the .45 is not significant. Notice in this example how the input and output quantities correspond to the input and output quantities of CR3 instead of to the input and output quantities of CR4 from which the division operator comes. We exploit the binding specifications of the correct rule to determine which quantities are the inputs and which is the output, because the goal is to compute the quantity the correct rule computes. Table 5.2 gives the number of occurrences of erroneous solution attempts in each category for students employing the operator from the derived/original rule.

<table>
<thead>
<tr>
<th>No incorrect quantities employed</th>
<th>19</th>
</tr>
</thead>
<tbody>
<tr>
<td>One incorrect quantity employed</td>
<td></td>
</tr>
<tr>
<td>bound to an input</td>
<td>9</td>
</tr>
<tr>
<td>bound to the output</td>
<td>1</td>
</tr>
<tr>
<td>Two incorrect quantities employed</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.2. Errors when employing the operator from a derived/original rule.

The hypothesis that students employ simplifications of rules explains these errors also. If we consider the selection of the proper operation to be a condition on the proper use of a rule, then a simplification of the rule eliminating this condition will allow selection of the wrong operation. If we constrain the simplifications to incorrectly choosing only one part of the rule (where part is defined to be a quantity or an operation), then we can explain the 16 errors when the correct rule (operation) is applied and the first 19 errors shown in Table 5.2. This constrained simplification hypothesis allows 35 of the 45 errors in the data to be explained. If we relax the constraint slightly to allow at most one operation and one quantity to be chosen, then all
of the errors are explained. We characterize this type of discrepant behavior as the incorrect application of correct rules, because the operation performed by the student so closely approximates the correct operation.

To see that there are conditions on the selection of an operator, consider how an experienced problem solver checks whether the units work out correctly in the solution attempted for the second problem in Figure 5.2 given above. To check his work, the experienced problem solver will add units to the expression to yield

\[ a \text{ gallons alcohol} = \frac{200 \text{ gallons solution}}{.45} \]

which is not a complete expression of the units. Next he will replace the .45 with an equivalent expression containing units and yielding

\[ a \text{ gallons alcohol} = \frac{200 \text{ gallons solution}}{45 \text{ gallons alcohol}} \cdot \frac{100 \text{ gallons solution}}{100 \text{ gallons solution}} \]

Because the right-hand side of this expression yields the units

\[ \frac{\text{gallons solution}^2}{\text{gallons alcohol}} \]

the condition on the selection of the proper operator has not been met. The simplified rule that the student applied to the problem does not take into account conditions on the proper use of operators. Because we do not include self-monitoring skills (such as those discussed above) in our model, al-Khorezmi does not uncover this error by employing self-monitoring. Instead, al-Khorezmi checks the student's expression by following the derivation/origination link from the correct rule to get the operator that the student may have employed. Adding the self-monitoring skills to our model
will enhance the ability to give corrective feedback on errors and the ability to teach more "expert-like" behavior.  

5.2.2. **Errors in forming and applying combined rules in wet-mixture problems**

Interpretation of a student's intent when he applies a combination of domain computation rules to produce a formula is more difficult. The complexity of the expressions adds to the difficulty of the analysis. From an analysis of the errors found in our data concerning the application of combined rules, we categorized the errors as either

- binding the lowest level expressions to illegal inputs, invalidating their output,
- equating a combined expression to a single quantity of the wrong type,
- equating expressions which compute different quantities, or
- employing a subexpression that is not a correct component of the operator applied to its result in the expression.

There were fifty-two errors involving binding the lowest level expression to illegal inputs. These errors correspond to the errors in binding the inputs of rules when solving problems requiring only a single rule application. We handle these errors in the manner discussed in the previous section, that is we match the operands of the student's operation against the inputs that the database specifies for computation rules that employ that operation. For example, on the problem

A fertilizer mixture containing 32% nitrogen is desired for a certain crop. The farmer has fertilizer with a 17% nitrogen content and another fertilizer with 43% nitrogen content. How

---

31 This has been just one example of the many forms of self-monitoring behavior that could be added. When we started this project, we explicitly omitted modeling self-monitoring behavior to concentrate effort on other areas. Therefore, we do not pursue this topic further here.
much of each fertilizer is needed to make 100kg of fertilizer with the desired 32% nitrogen content? (Phillips et al., 1983, p. 148)

One student attempted to apply the expression

\[ 0.17x + 43y = 100 - 0.32. \]

In this expression the operands of the subtraction (the amount of the fertilizer mixture and the proportion of nitrogen in the fertilizer mixture) do not match the inputs to CR11 (the amount of the fertilizer mixture and the amount of the component of the fertilizer that is not nitrogen) nor do they match the inputs to CR7 (the full proportion constant 1.0 and the proportion of nitrogen in the fertilizer mixture).

There were eight errors in which a single expression was equated to the wrong quantity and no errors in which two expressions computing different quantities were equated. We predict errors in which two expressions computing different quantities are equated will happen, although such errors did not appear in our data from wet-mixture problems. These errors violate the "computation" rule for the equals operator. This rule says one may only equate two "equivalent" inputs, that is they are the same quantity or they are related to each other by an identity relationship in the problem. Therefore, the occurrence of these errors appears to be caused by the simplification of the conditions of applicability of the equals operator. For example, on the problem given above one of the students wrote the equation

\[ 0.17x + 0.43y = 0.32. \]

The expression on the left-hand side computes the amount of nitrogen in the mixture whereas the value on the right-hand side is the proportion of nitrogen in the mixture. In such a situation, the student must be careful to equate equivalent quantities.

Our evaluation of test answers that applied combinations of rules revealed eight occurrences of employing a subexpression that is not a correct
component of the operator applied to its result in the expression. Because the input of a rule employs the output of another rule, it is more difficult to attribute the error to one or the other. For example, a student attempted to solve the problem given above with the expression \[ 0.17 + 0.43x = 32 \times 100. \]

Here it is correct to multiply \(0.43\) times \(x\) when applying CR3 to produce the amount of nitrogen in the second fertilizer, but the inputs, \(0.17\) and \(0.43x\), to the addition operator are not correct for any of the computation rules. Therefore, if our data contained an error where an operator had the output of a subexpression as its input, then we assumed that the subexpression computed the proper output and attributed the error to employing incorrect input quantities for the higher level operator. In the above example we attributed the error to the incorrect inputs to the addition operator, not to the multiplication of \(0.43\) times \(x\). This approach is consistent with our treatment of the other situations; when an error occurred at the lowest level, we attributed it to the student providing incorrect inputs and when an error occurred in equating quantities, we did the same.

A more sophisticated approach would be to employ the student model to compare the student's knowledge of the rules with his propensity to employ deviations corresponding to either error. By propensity to employ deviations corresponding to either error, we mean the student's history of misapplying a generalized computation rule and not the student's history of applying particular specializations. In the above example this means comparing the propensity of the student to err by adding the proportion of nitrogen in the first fertilizer (\(0.17\)) to the amount nitrogen in the second fertilizer (\(0.43x\)) to the propensity of the student to err by multiplying the proportion of nitrogen in the second fertilizer (\(0.43\)) times the amount of the second fertilizer (\(x\)) to
produce the proportion of the first fertilizer which is not nitrogen (the only quantity that can be legally added to the .17). The system could then draw a conclusion from this comparison about which error the student made.

To get the student to learn the proper combination of expressions, he should be taught the strategic rules that will lead to the formation such combinations. Specifically, strategic rule ME4 makes unknown input quantities into subgoals and strategic rule ME2.1 chooses rules with subgoals as their output (the strategic rules discussed here are repeated in Figure 5.3 for convenient reference). In the example above the student is attempting to complete

\[
\text{amount nitrogen in first fertilizer} + \text{amount nitrogen in second fertilizer} = \text{amount nitrogen in desired mixture} \ (0.32 \cdot 100).
\]

Strategic rule ME4 would make the first two amounts subgoals and rule ME2.1 would choose computation rule CR3 to produce the expressions

\[
\text{amount nitrogen in first fertilizer} = 0.17y
\]

and

\[
\text{amount nitrogen in second fertilizer} = 0.43x
\]

where \(y\) and \(x\) are the amount of the first fertilizer and the amount of the second fertilizer respectively. Applying strategic rule ME3.2 to these expressions yields the desired equation

\[
0.17y + 0.43x = 0.32 \cdot 100.
\]

---

32 This is not done by a model-tracing tutor such as ours. Appendix D addresses the issues of how and when new strategic rules such as ME4 and ME2.1 should be taught.
ME2.1: Given a goal attribute then select a rule computing its value as output and determine consistent bindings.

ME3.2: Given two equations having a nonprimary goal variable in common then combine equations eliminating the nonprimary goal variable.

ME4: Given a bound computation rule with unknown nongoal attributes and an unknown goal attribute then make the nongoal attributes be goal attributes.

Figure 5.3. Strategic rules ME2.1, ME3.2, and ME4.

5.2.3. **Mistaken analogies in wet-mixture problems**

In the discussion above, we attributed observed errors to the misapplication of correct rules by simplification of the constraints for binding quantities or selecting operators. We have not discussed the incorrect rule application that we observed. For the wet-mixture problems, we found students applied two incorrect base rules, DCR1 and DCR3 given in Figure 5.4 with CR8. Summing the proportions of a substance in two component solutions to give the proportion of the substance in the composite solution (DCR1) is the most common incorrect rule they applied. A mistaken analogy to CR8 forms this rule. There is a direct mapping from the binding of the three amounts of substances in CR8 to the binding of the proportions of substances in the deviant rule. We consider a mistaken analogy such as this one to be an extreme form of oversimplification of CR8. In this oversimplification all three quantities are bound incorrectly. Our data from the test papers showed students applying DCR1 four times in single rule situations, and five times in combined rule situations. Because this simplification is so much different than those labeled as incorrect applications of correct rules, we represent it as
a separate, *deviant*, rule that students hold. Further evidence that these rules are incorrect rules that students hold and apply correctly comes from the degree to which they defend their application. For example, during the experimental use of al-Khorezmi, student RL (see Appendix B for the record of the tutorial interaction) adamantly insisted that al-Khorezmi was wrong when it pointed out that he had made a mistake by applying DCR1. In contrast, no student objected when al-Khorezmi informed them they had incorrectly applied a correct rule.

Besides the base rules DCR1 and DCR3, DCR2, the derived form of rule DCR1, occurred once in a single rule situation and once in a combined rule.
CR8: The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating solutions.  
Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively  
Constraints: C2 not equal C3  
Input values: amount(S2), amount(S3)  
Output value: amount(S1)  
Computation: amount(S2) + amount(S3)

DCR1: The proportion of a substance in the composite solution is equal to the sum of the proportions of like substances in the participating solutions.  
Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively  
Constraints: C2 not equal C3  
Input values: proportion(S2), proportion(S3)  
Output value: proportion(S1)  
Computation: proportion(S2) + proportion(S3)

DCR2: The proportion of a substance in a participating solution is equal to the difference between the proportion of the like substance in the composite solution and the proportion of the like substance in the other participating solution.  
Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively  
Constraints: C2 not equal C3  
Input values: proportion(S1), proportion(S3)  
Output value: proportion(S2)  
Computation: proportion(S1) - proportion(S3)

DCR3: The proportion of a substance in the composite solution is equal to the average of the proportions of like substances in the participating solutions.  
Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively  
Constraints: C2 not equal C3  
Input values: proportion(S2), proportion(S3)  
Output value: proportion(S1)  
Computation: \[
\frac{\text{proportion}(S2) + \text{proportion}(S3)}{2} + 2
\]

Figure 5.4. Computation rules CR8, DCR1, DCR2, and DCR3.

In addition to finding that students apply deviant rules "correctly," we found that the students also apply deviant rules incorrectly. The data from
student test papers show instances of students applying these deviant rules with invalid bindings of inputs or with operators from derived forms of the deviant rules. These incorrect applications are similar to those in which students incorrectly applied correct rules. For example, in attempting to solve

A fertilizer mixture containing 32% nitrogen is desired for a certain crop. The farmer has fertilizer with a 17% nitrogen content and another fertilizer with 43% nitrogen content. How much of each fertilizer is needed to make 100kg of fertilizer with the desired 32% nitrogen content? (Phillips et al., 1983, p. 148)

one student wrote the expression

\[ 32x + (43 - 17) = 100. \]

Here the attempt to subtract the two percentages (or proportions) from two different mixtures is not only a deviant operation but is also in contradiction with the "proper" binding of DCR2 which would give 32 - 17 as the computation. Therefore, incorrect application of deviant rules can be uncovered by applying to the representations of the deviant rules the same techniques exploited to uncover incorrect application of correct rules. For example, we discussed earlier that one way an incorrect application of a correct rule is uncovered is to check whether the rule matches all of the students operation except the binding of one input. The same match can be performed for the deviant rule which would uncover an incorrect application of an incorrect rule.

The other, less common deviant rule, DCR3 is to divide by two the sum of the proportions of the same substance in the two combined solutions to get the proportion of that substance in the composite solution. The application of this rule occurs two times in the data. We propose that this is a mistaken analogy to other domains, such as test grades, where percents or proportions are summed and averaged.
5.2.4. Errors in solving distance-rate-time problems

In addition to the data concerning solving wet-mixture problems that we collected and studied, we have a smaller amount of data from student attempts to solve distance-rate-time problems. Because all the students that attempted simple problems on the tests that we gave correctly solved the problems, we have only error data from test problems that required computation rule combinations. In this data, there were fourteen errors involving binding the lowest level expression to illegal inputs. There was one error in which a single expression was equated to the wrong quantity, no errors in which two expressions computing different quantities were equated, and no occurrences of employing a subexpression that is not a correct component of the operator applied to its result in the expression. Because the category for binding the lowest level expression to illegal inputs provides an explanation for any illegal use of an operator, we can fit all low level errors into it. The other categories catch the remaining errors after the lowest level expressions have been correctly constructed. This approach emphasizes the importance of associating the proper operands with their operators. This is not to say that this approach always gives the correct explanation, but that it gives a feasible explanation. For example, if a student writes the expression

\[ \frac{400}{r} - 20 = \frac{500}{r} + 20 \]

for the problem

Beth can fly her plane 400 kilometers against a 20 kph wind and 560 kilometers with that wind in the same amount of time. What is the speed of the plane?

then according to our approach the errors are dividing the 400 and the 500 by \( r \), whereas the correct explanation is that the expression should be written
The protocol data and our experience in tutoring students showed mistaken analogies similar to the summing of proportions across solutions. In problems involving a single person or object undertaking two separate instances of travel, some students summed the rates given to get a "total rate" and others summed the rates and divided by two to get the average rate. The correspondence of these two deviations to the deviations of summing and averaging proportions in wet-mixture problems supports our analysis. Furthermore, the rate in

\[
\frac{400}{r - 20} = \frac{500}{r + 20}
\]

distance = rate \times time

of distance-rate-time problems plays the same role as the proportion does in CR3 of the wet-mixture problems and these rules were shown to be analogous in Chapter 4.

5.2.5. Comparison to other work

Our primary concern is the explanation of mistakes by their genesis. The previous sections showed that simplification and mistaken analogy explain a high percentage of the mistakes we observed. Matz (1982) and Brown and VanLehn (1980) develop models that predict bug occurrence from an analysis of bugs. Matz accounts for the errors by proposing that solvers employ extrapolation techniques on their base rules to apply a known rule inappropriately in a new situation, or incorrectly adapt a known rule to a new problem. She gives two extrapolation techniques, linearity and generalization.
Linearity errors occur from applying an operator linearly to each subpart of an object and combining the subparts. *Generalized distribution* and *repeated application* are types of linearity errors.

An example of generalized distribution is

$$\sqrt{A + B} \Rightarrow \sqrt{A} + \sqrt{B}$$

where the radical sign is distributed over the A + B. An example of repeated application is

$$\frac{A \cdot X + B \cdot Y}{X + Y} \Rightarrow A + B$$

where the base rule

$$\frac{A \cdot X}{X} \Rightarrow A$$

is repeatedly applied. Loosening the conditions of applicability (simplification) of a known rule is an interpretation of the genesis of these errors. The student does not have all the refinements (conditions of applicability) available and so goes ahead with the operation when inappropriate. Matz says

"although linear assumptions are at times inappropriate, we cannot say that linear extrapolation techniques are inherently incorrect. Rather, their misuse stems both from not recognizing that such an assumption has been made, and from an inadequate knowledge of *semantic constraints."* (p. 33, emphasis added)

Generalizations revise a known rule to accommodate the situation. This accommodation is exactly the removal of constraints on the applicability of the rule. Therefore, Matz's term generalization is the same as Goldstein's simplification. An example is generalizing the solution of

$$(X - A)(X - B) = 0$$

as

$$X - A = 0, \text{ or } X - B = 0$$
\( (X - B) = 0 \)

to

\( (X - A)(X - B) = K \)

as

\( (X - A) = K, \) or

\( (X - B) = K. \)

Matz reports that experts are often observed checking the applicability of rules that are illegal by substituting in values. This is interesting because it implies that even experts do not retain in their memory all the conditions of applicability of a rule. They make up for storing simplified rules by employing checks. This suggests that modeling expert competence includes modeling the checks an expert makes. Checking that the units come out correct is a primary check for correct bindings of quantities in rules in algebra word problems.

Brown and VanLehn (1980) propose a generative theory of bugs which they exploit to explain the creation of bugs during a student's learning experience. Their work corresponds to Matz's in proposing that students develop bugs when they have incomplete knowledge of how to proceed and an impasse occurs. They propose general repair techniques with repairs being filtered out by critics. They further note that preconditions of primitive operations filter repairs just as critics do.

To exploit this tie between critics and preconditions further, we suggest that application of rules without all of their preconditions available causes the need for critics. Matz shows experts employ critics by reporting data showing experts applying a rule when full preconditions would preclude its application and critiquing it by investigating examples. However, Brown and VanLehn proceed differently by employing rule deletion--constrained by not allowing
simplifications to be deleted if there are still refinements in the rule set--to create impasses. Because the preconditions of other rules prevent them from being applied, an impasse occurs when a rule is applicable, but deleted.

The concept of rule simplification explains many of the general purpose repair heuristics of Brown and VanLehn. For example, they propose a *swap vertically* repair heuristic which swaps the order of the digits in a subtraction problem. This repair is directly analogous to reversing the order of the operands in noncommutative operations that we observed in our data. When a student need only apply a correct rule to the operands he is employing and he employs the operands in reverse order, it is difficult to conclude that he reached an impasse and made a repair. Instead, at least in this case, employment of a simplification caused by a lack of proper preconditions is a better explanation.

Our interest in errors in this thesis is primarily one of cataloging and explaining. We realize that better theories for prediction of errors will lead to more capable intelligent tutoring systems and propose further work be done on the application of simplification and mistaken analogy to predict errors.

Venezky and Bregar (1988) support modeling of expert knowledge for solving word problems as the incorporation of critiquing techniques. They report that more experienced problem solvers differ considerably from less experienced problem solvers in control and monitoring strategies. The more experienced solvers check their work whereas the inexperienced solvers' work is nearly devoid of checks. Wollman (1983) reports that a method for checking correctness or plausibility is the crucial step in correctly solving
problems like the students and professors problem. But, since we represent deviations of rules, al-Khorezmi matches occurrences of such behavior and provides explanation of the mistake with a suggestion of how checking prevents the error. There are other problems where the wording of the problem strongly suggests an incorrect formulation. An example is:

Tom drove 240 miles the first day at 40 mph and 300 miles the second day at 60 mph. What was his average rate of speed while driving?

For this problem, students typically average the speed with \( \frac{40 + 60}{2} \).

Collins et al., (1987) propose cognitive apprenticeship as a new approach to teaching because traditional schooling has left conceptual and problem-solving knowledge largely unintegrated or inert for many students. Cognitive apprenticeship teaches the methods experts exploit to handle complex tasks. This requires formulating explicit models of expertise to be taught and encouraging "the development of self-correction and monitoring skills" (p. 5). Therefore, we propose that future models for intelligent tutoring systems incorporate a model of correction and monitoring skills.

5.3. Uncovering mistakes through model tracing

Chapters 2 and 3 discuss our employment of the model-tracing paradigm to construct al-Khorezmi. Employing model tracing requires al-Khorezmi to trace every operation the student makes and provide corrective feedback for the student when errors are made. In the following sections we discuss the two menu operations that indicate the student made a qualitative processing step. Because we employ model tracing, al-Khorezmi evaluates the step the

33We do not provide a model for translation from English of problems like the students and professors problem (translate into an equation--there are 6 times as many students as professors).
student takes and if it finds the step to be in error it provides appropriate corrective feedback.

5.3.1. **Give-calculated-result option errors**

When a student employs the give-calculated-result option, al-Khorezmi assumes that he has bound one rule and fired it (strategy rules FD2.1 and FD3.1) to obtain his result. Therefore, al-Khorezmi examines the rules which calculate the quantity for which the student gave his result, fire any that apply to the current problem, and compare its results to the students. If no rule computations match the student's result, then al-Khorezmi considers whether the student has made any of the errors in our categorization presented in the previous section. Upon finding an error corresponding to those discussed in the previous section, al-Khorezmi displays the error with its correction to aid the student in understanding his mistake.

5.3.2. **Give-expression option errors.**

When the student selects this option he types in an expression that he feels properly relates the quantities in the problem. Al-Khorezmi then evaluates how each operation in the student's expression relates quantities to form values for new quantities. If all relationships formed by operators between related operands are valid, it announces that the expression is valid. However, if there are one or more invalid relationships, then al-Khorezmi employs the error categorization of the previous section to determine how each relationship is invalid after which it prints the proper correction. Noting three or more errors in combining terms is possible when a formula has complex expressions on both sides of the equals sign. For example, on a test one student wrote the expression
where \( A_i \) is the amount of solution \( i \), and \( p_{ij} \) is the proportion of substance \( j \) in solution \( i \). The three division operations relate inappropriate quantities. Therefore, to relate appropriate quantities with the division operation, al-Khorezmi says to replace each amount of solution \( i \) with the amount of substance \( j \) in solution \( i \) giving the expression

\[
\frac{a_{j1}}{p_{11}} + \frac{a_{j2}}{p_{21}} = \frac{a_{j3}}{p_{31}}
\]

(\( a_{ij} \) is the amount of substance \( j \) in solution \( i \)) which is equivalent to

\[ A_1 + A_2 = A_3. \]

5.4. Representing erroneous behavior

In section 5.2 we categorized the errors found in data collected from student problem solvers. Our evaluation of the data found uniform error types occurring in the application of computation rules from different domains. This suggests not representing every deviation to every rule with a static representation to form a bug catalog. Instead, we represent the correct rules with a static representation and the binding problems with dynamic processes. The static representation contains a generalized description of the structure of the quantities to be bound by the rule. The description is generalized because often there are several different instantiations of a rule possible in a problem type. Al-Khorezmi applies unification to this generalized rule description and the description of the students expression to determine the particular instantiation employed by the student. For example, at \( \{2\} \) in Figure 5.6 the student encodes the expression

\[ c + i = q \times o \]
which stands for the proportion of the alcohol in the 20% solution plus the amount of alcohol in the pure alcohol equals the proportion of alcohol in the 70% alcohol solution times the amount of the 70% alcohol solution. The right-hand side is represented as

\[(\text{solution3 substance1 proportion}) \times (\text{solution3 amount})\]

which unifies with the representation of the computation for CR3,

\[(\text{solutionx substancey proportion}) \times (\text{solutionx amount}),\]

by employing the substitutions \text{solution3} for \text{solutionx} and \text{substance1} for \text{substancey}. The left-hand side of the expression does not unify with any computation rule for the domain. If the binding employed by the student is inconsistent with any of the correct bindings for the rule then al-Khorezmi assumes the student has simplified (the genetic graph simplification relationship) the conditions for binding the quantities to the rule. The other possibility of an error is that the student simplified the conditions for choosing an operator to apply to the quantities employed. Al-Khorezmi checks for this possibility by applying a procedure that checks for the employment of an operator from a derived (original) rule by following derivation (origination) links.

We represent deviant rules, such as DCR1 in which proportions are summed across solutions in wet-mixture problems, with a static representation instead of with a procedure which checks for the error possibilities categorized in the previous section. We do this because we consider these deviations to be incorrect rules applied correctly. Therefore, if al-Khorezmi considers the possibility that the student applied a deviant rule and the deviant rule had more than one instantiation in the problem, then al-Khorezmi employs unification to determine which instantiation the student may have made. For example, at \(I\) in Figure 5.6 the student encodes the expression
\( c + j = q \)

which stands for the proportion of the alcohol in the 20% solution plus the proportion of alcohol in the pure alcohol equals the proportion of alcohol in the 70% alcohol solution. This expression is represented as

\[
(s_{\text{solution1}} \text{ substance1 proportion}) + (s_{\text{solution2}} \text{ substance1 proportion}) = (s_{\text{solution3}} \text{ substance1 proportion})
\]

which unifies with the representation of the computation for DCR1,

\[
(s_{\text{solution1}} \text{ substancex proportion}) + (s_{\text{solution2}} \text{ substancex proportion}) = (s_{\text{solution3}} \text{ substancex proportion}),
\]

by employing the substitution \text{substance1} for \text{substancex}. The is as opposed to the other "correct" instantiation of deviant rule DCR1 in which \text{substancex} is replaced by \text{substance2}.

Finally, we observed in our data that students apply deviant rules incorrectly in the same manner that they apply correct rules incorrectly by employing the operator from the derived/original rule or by binding an inappropriate quantity to the rule. Therefore, employing the dynamic processes created to find incorrect applications of correct rules on the descriptions of deviant rules enables discovery of incorrect applications of incorrect rules.

5.4.1. Examples of corrective feedback from al-Khorezmi

In our experimental use of al-Khorezmi with twelve students from Lutheran High School, seven students JS, JH, JW, RL, KL, BM, and CW made errors in applying domain rules. All of their errors were in the categories that we derived from our analysis of data collected from tests. Therefore, al-Khorezmi found and provided corrective feedback for all of their errors. Five students applied either the deviant rule for summing proportions across solutions or its derived form. One student twice bound an inappropriate
quantity into rule instantiations. And, two students bound inappropriate quantities into combined rules.

5.4.2. **Corrective feedback from the give-calculated-result option**

Figure 5.5 shows a student making mistakes in binding an input quantity into a computation rule. At (1) he divides the 24 gallons of water in the mixture into the 6 gallons of acid in an attempt to obtain the proportion of acid. Al-Khorezmi tells him that he needs to divide by the total amount of the mixture instead. Furthermore the interaction shows al-Khorezmi trying to explain that he needs to be more careful (refine) in his application of the rule, making sure to select the correct entity attribute for binding to a quantity in the rule. At (2) he multiplies the proportion .66 of copper in the 66% copper alloy times the 80 pounds of the second alloy in an attempt to obtain the amount of the weight of the copper. Al-Khorezmi gives corrective feedback similar to that given for his first mistake.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

<table>
<thead>
<tr>
<th>acid</th>
<th>water</th>
<th>amount-30%-solution</th>
<th>amount-15%-solution</th>
<th>amount-mixture</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>a)</td>
<td>b)</td>
<td>c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d)</td>
<td>e)</td>
<td>f)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>g)</td>
<td>h)</td>
<td>i)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>j)</td>
<td>k)</td>
<td>l)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>m)</td>
<td>n)</td>
<td>o)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>p)</td>
<td>q)</td>
<td>r)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>s)</td>
<td>t)</td>
<td>u)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>v)</td>
<td>w)</td>
<td>x)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>y)</td>
<td>z)</td>
<td></td>
</tr>
</tbody>
</table>

1) note goal 5) give expression 9) help-algebra
2) note constant 6) solve 10) exit-problem
3) specify a variable 7) help-hints
4) give calculated result 8) help-menu-options

Option? 4 Quantity? b Value? 3
OPERATION VERIFIED TO BE CORRECT

Installing problem constant 0.3 in proportion-alcohol-30%-sol as you made use of it in computing your result. Installing problem constant 10.0 in amount-30%-solution as you made use of it in computing your result.

Option? 4 Quantity? e Value? 7
OPERATION VERIFIED TO BE CORRECT

Option? 4 Quantity? i Value? 3
OPERATION VERIFIED TO BE CORRECT

Installing problem constant 0.15 in proportion-alcohol-15%-sol as you made use of it in computing your result. Installing problem constant 20.0 in amount-15%-solution as you made use of it in computing your result.

Option? 4 Quantity? l Value? 17
OPERATION VERIFIED TO BE CORRECT

Option? 4 Quantity? p Value? 6
OPERATION VERIFIED TO BE CORRECT

Option? 4 Quantity? s Value? 24
OPERATION VERIFIED TO BE CORRECT

Figure 5.5. A portion of BM's problem-solving interaction.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

```
a) amount-30%-solution 10.0
 acid  
 water
 b) amount 3
c) proportion 0.3
d) percent
 e) amount 7
 f) proportion
g) percent
 h) amount-15%-solution 20.0
 acid  
 water
 i) amount 3
 j) proportion 0.15
 k) percent
 l) amount 17
 m) proportion
 n) percent
 o) amount-mixture
 acid  
 water
 p) amount 6
 q) proportion 0.2
 r) percent
 s) amount 24
 t) proportion
 u) percent
 v) full proportion constant 1.0
 w) full percent constant 100.0
 x) other value
```

1) note goal 5) give expression 9) help-algebra
2) note constant 6) solve 10) exit-problem
3) specify a variable 7) help-hints
4) give calculated result 8) help-menu-options

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason? 1

Would you like to 1) stop now 2) load a new problem? 2

Figure 5.5. (continued).
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

| pure-copper | b) amount | c) proportion | d) percent |
| other-metal | e) amount | f) proportion | g) percent |
| pure-copper | i) amount | j) proportion | k) percent |
| other-metal | l) amount | m) proportion | n) percent |
| pure-copper | p) amount | q) proportion | r) percent |
| other-metal | s) amount | t) proportion | u) percent |

v) full proportion constant 1.0  
w) full percent constant 100.0  
y) other value

1) note goal  
2) note constant  
3) specify a variable  
4) give calculated result  
5) give expression  
6) solve  
7) help-hints  
8) help-menu-options  
9) help-algebra  
10) exit-problem

That value doesn't belong with any quantity in this problem. You used a simplified version of the rule: THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION. You must refine your selection of quantities to amount-66%-alloy instead of amount-second-alloy.

Figure 5.5. (continued).

5.4.3. Corrective feedback from the give-expression option

Figure 5.6 shows a student applying a deviant rule \(\{1\}\) for summing proportions across solutions. Al-Khorezmi corrects her and tells her how to check for the reasonableness of the rule. At \(\{2\}\) she sets up a formula with one of its input quantities bound incorrectly, at which time al-Khorezmi corrects her and suggests that she refine her knowledge of applying the rule that she attempted to employ in the formula.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[ \text{alcohol} \quad \text{b)} \quad \text{amount} \quad \text{c)} \quad \text{proportion} \quad \text{d)} \quad \text{percent} \]
\[ \text{water} \quad \text{e)} \quad \text{amount} \quad \text{f)} \quad \text{proportion} \quad \text{g)} \quad \text{percent} \]

\[ \text{alcohol} \quad \text{h)} \quad \text{amount-pure-solution} \quad \text{i)} \quad \text{amount} \quad \text{j)} \quad \text{proportion} \quad \text{k)} \quad \text{percent} \]
\[ \text{water} \quad \text{l)} \quad \text{amount} \quad \text{m)} \quad \text{proportion} \quad \text{n)} \quad \text{percent} \]

\[ \text{alcohol} \quad \text{o)} \quad \text{amount-70%-solution} \quad \text{p)} \quad \text{amount} \quad \text{q)} \quad \text{proportion} \quad \text{r)} \quad \text{percent} \]
\[ \text{water} \quad \text{s)} \quad \text{amount} \quad \text{t)} \quad \text{proportion} \quad \text{u)} \quad \text{percent} \]

v) full proportion constant 1.0 \quad w) full percent constant 100.0 \quad y) other value

1) note goal \quad 5) give expression \quad 9) help-algebra
2) note constant \quad 6) solve \quad 10) exit-problem
3) specify a variable \quad 7) help-hints
4) give calculated result \quad 8) help-menu-options

\[ \text{Option? 5 Expression? } \text{c+j=q} \quad \text{Hit return to continue.} \]

Menu option c is as yet unspecified. Please enter a value or variable for it: 2
Menu option j is as yet unspecified. Please enter a value or variable for it: x 1
Menu option q is as yet unspecified. Please enter a value or variable for it: 0.7

The following equation was not found to be valid: \[ (0.2 + x_1) = 0.7 \]

You used the following ILLEGAL rule: WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE PROPORTION OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING THE PROPORTIONS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS to combine 0.2 of proportion-alcohol-20%-sol with x 1 of proportion-alcohol-pure-sol.

This rule is wrong because: Summing the amounts of these substances is legal (cr8) but it is a mistake to make an analogy and sum the proportions of these substances. You should monitor your use of computational relationships. When two solutions are mixed, the proportion of a substance in the solution falls between the proportions of the substance in each of the solutions.

Figure 5.6. A portion of CW's problem-solving interaction.
Menu option i is as yet unspecified. Please enter a value or variable for it: x2
Menu option o is as yet unspecified. Please enter a value or variable for it: x3

amount-alcohol-70%-sol does not combine with left operand using equate.

You used a simplified version of the rule: WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS. You must refine your selection of quantities to amount-alcohol-20%-sol instead of proportion-alcohol-20%-sol.

Menu option b is as yet unspecified. Please enter a value or variable for it: 2

The following equation was not found to be valid: (0.2 + x2) = (0.7 * x3)

Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

EXPRESSION VERIFIED TO BE CORRECT

\[ ((2 + x2) / x3) = 0.7 \]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

1) note goal 5) give expression 9) help-algebra
2) note constant 6) solve 10) exit-problem
3) specify a variable 7) help-hints 4) give calculated result 8) help-menu-options
6. Implementation details

In this chapter we discuss significant facets of the implementation of al-Khorezmi including:

- making the strategic rules independent of the computation rules for a particular domain;
- keeping a record of a student's capability and making use of it to provide tutorial information; and
- making use of the frontier of a student's knowledge in the genetic graph.

This discussion also covers our employment of the Flavors (Weinreb and Moon, 1981; Allen et al., 1983) object-oriented programming package of YAPS (Allen, 1983) to provide structured data types representing entities of the domain, generalized processing of these types, and sharing of operations between different flavors.

6.1. Strategic rule independence from domain computation rules

Our goal is to assist students in learning flexible problem-solving techniques, as opposed to learning domain specific problem-solving techniques. We argue that the significance of a single set of strategic rules lies in the fact that they are general and powerful enough to work in the context of different domains. Because our implementation of this domain independent set of strategic rules is general, it can manipulate the different quantities and computational relationships (encoded in schemas and computation rules respectively) of these different domains. Therefore, we have the flexibility necessary to reach our goal of teaching flexible problem-solving techniques.

We use the Flavors system in conjunction with a unification process to provide the desired flexibility. We represent the various domain schemas
using flavors objects. For example, the schema for wet-mixture problems is represented with the flavor definitions given in Figure 6.1, and the schema for distance-rate-time problems is represented with the flavor definitions in Figure 6.2. The advantage of using Flavors here comes from the ability to use a *variable component name*, as opposed to a *literal component name*, for selection of the desired *instance variable*.\(^{34}\) We include instance variable names in the representation so that al-Khorezmi can bind them to variables and employ these variables to select, from Flavors objects, the values of the instance variables named. This capability of selecting components of a data structure using the component name bound to a variable would require a complicated conditional selection process be coded in other programming languages, whereas it is built into the processing of message passing in Flavors. The effect is to enable the use of a variable to select components of heterogeneous composite types like record structures in a manner similar to the way most programming languages use variables to select components of homogeneous composite types or arrays.

---

\(^{34}\)Instance variables are components of a heterogeneous structure that can be selected by their names.
(defflavor amount ; Flavors amount, proportion, and percent represent the basic ; quantities from which the schema is built.
  ((amount-value (gensym 'x))) ; Initially an unknown variable.
  (amount-units *UNBOUND*)
  (amount-pname *UNBOUND*)
  (utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor proportion
  ((proportion-value (gensym 'x))
  (proportion-pname *UNBOUND*))
  (utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor percent
  ((percent-value (gensym 'x))
  (percent-pname *UNBOUND*))
  (utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor substance ; A substance has an amount and
  ((parent nil)) ; a proportion and percent of some parent entity.
  (instance *UNBOUND*) ; Holds the instance variable name that selects it.
  (name type amount proportion percent goal utility db-utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor solution-mix ; A solution mixture has an amount, and is made of
  ((parent nil)) ; two substances and may participate in a composite
  (instance *UNBOUND*) ; mixture relation.
  (first-substance (make-instance 'substance))
  (second-substance (make-instance 'substance))
  (name amount goal utility db-utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor mixture-relation ; A composite mixture relationship is made up of three
  ((parent nil)) ; solutions.
  (instance *UNBOUND*)
  (first-solution (make-instance 'solution-mix))
  (second-solution (make-instance 'solution-mix))
  (composite-solution (make-instance 'solution-mix))
  (full-proportion-constant 1.0) ; Constant values used in solving problems which use
  (full-percent-constant 100.0) ; this schema.
  (other-value *UNBOUND*)
  (accesses (make-instance 'mixture-access-paths))
  (name utility db-utility)
  :gettable-instance-variables
  :settable-instance-variables)

Figure 6.1. Flavors for wet-mixture problem schema.
(defflavor distance ; Flavors distance, time, and rate represent the basic quantities ; from which the schema is built.
   ((distance-value (gensym 'd)) (distance-units 'miles)(distance-pname '*UNBOUND*))
   (utility)
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor time
   ((time-value (gensym 't)) (time-units 'hours)(time-pname '*UNBOUND*))
   (utility)
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor rate
   ((rate-value (gensym 'r)) (rate-units 'miles/hour)(rate-pname '*UNBOUND*))
   (utility)
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor moving-object ; A moving object has a rate and moves a certain distance
   ((parent nil) ; in a specific time.
   (instance *UNBOUND*))
   (name type distance rate time goal utility db-utility))

(defflavor opposite-direction-distance-relation
   ((parent nil) ; Two objects moving in opposite directions.
   (instance *UNBOUND*)
   (first-object (make-instance 'moving-object))
   (second-object (make-instance 'moving-object)))
   (name distance goal utility db-utility)
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor same-direction-distance-relation
   ((parent nil) ; Two objects moving in the same direction.
   (instance *UNBOUND*)
   (first-object (make-instance 'moving-object))
   (second-object (make-instance 'moving-object)))
   (name distance goal utility db-utility)
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor two-object-rate-relation
   ((parent nil) ; Two objects moving in the same direction at different
   (instance *UNBOUND*) ; rates. Rate is the difference between their rates.
   (slower-object (make-instance 'moving-object))
   (faster-object (make-instance 'moving-object)))
   (name rate goal utility db-utility)
   :gettable-instance-variables
   :settable-instance-variables)

Figure 6.2. Flavors for distance-rate-time problem schema.
(defflavor two-object-time-relation
  ((parent nil) ; Two objects moving for a different amount of time.
   (instance *UNBOUND*) ; Time is the difference between the times they travel.
   (slower-object (make-instance 'moving-object))
   (faster-object (make-instance 'moving-object)))
  (name time goal utility db-utility)
  :gettable-instance-variables
  :settable-instance-variables)

(defflavor in-moving-substance-relation
  ((parent nil) ; Two objects traveling in opposite directions in a
   (instance *UNBOUND*) ; moving substance such as water or air (wind).
   (with-substance (make-instance 'moving-object))
   (against-substance (make-instance 'moving-object))
   (moving-substance (make-instance 'moving-object)))
  (name rate goal utility db-utility)
  :gettable-instance-variables
  :settable-instance-variables)

Figure 6.2. (continued).

Not only can single instance variable names be bound to variables, but a multilevel schema, such as flavor mixture-relation given in Figure 6.1, can have its subordinate components selected by using a list of instance variable names to repeatedly select values. We call these lists of instance variable names access paths and apply functions access and access-next-to-last (given in Figure 6.3) to access instance variables through these access paths. The function access retrieves the value of an instance variable from a multilevel structure when given its access path. The function access-next-to-last accesses the environment in which the instance variable at the end of the access path exists so that the value of that instance variable can be set. Many objects can calculate their own access path, because many flavors mix in flavor, utility, which has a method make-access-path. Also, each object has an instance variable, parent, which stores a reference to the parent object of the object. There are parent objects because the schemas are formed from hierarchical structures in which flavor objects are stored subordinate to other
flavor objects. For example, when an instance of flavor mixture-relation (in Figure 6.1) is made, it has three subordinate objects of flavor solution-mix.

```
(defun access (val list) ; Use access path, list, to access thru val.
  (cond
    ((null list) val) ; On empty access path return current val.
    ((eq *UNBOUND* val) *UNBOUND*)
    (t (access (apply '<< (list val (car list))) (cdr list))))
)

(defun access-next-to-last (val list)
  (cond
    ((null (cdr list)) val) ; Return value when there is one remaining access.
    (t (access-next-to-last (apply '<< (list val (car list))) (cdr list))))
)
```

Figure 6.3. Functions access and access-next-to-last.

Component selection that employs variables bound to access paths, which contain instance variable names, enables the computational relationships of our computation rules to be described in a general manner. This is done by placing unknowns in the access paths to quantities that computation rules of a particular problem type schema bind. Placing unknowns in the access paths allows having only one description of a computation rule for a domain. The alternative is to have a description for each instance of the applicability of the computation rule. Figure 6.4 gives the LISP expression that generates the description of CR3 and asserts it into the database. Without using unknowns (generated using gensym) in access paths, CR3 would require six descriptions in the database.

The strategic rules determine which computation rules to apply by unifying the unknowns in the access paths of the description of the computation rule with the access paths of the quantities in the problem representation. Because this unification process operates on access paths of
any domain, the strategic rules are general and can operate on the computation rules and schema of any domain.

\[
; \text{CR3 multiplies the proportion of a substance (common-substance) in a solution} \\
; \text{(common-solution) times the amount of the solution (common-solution) to give the} \\
; \text{amount of the substance (common-substance) in the solution (common-solution).} \\
\text{(let ((common-solution (gensym 'x)))} \\
\text{(common-substance (gensym 'x)))} \\
\text{(fact rule-computation}} \\
\text{cr3} \\
\text{(times (\^\text{gensym 'x})} \\
\text{(\^common-solution \^common-substance proportion-value))} \\
\text{(\^\text{gensym 'x})} \\
\text{(\^common-solution amount-value))} \\
\text{(\^common-solution \^common-substance amount-value) nil))}
\]

Figure 6.4. LISP expression which generates the description of CR3.

6.2. Student modeling

This thesis emphasizes the representation of the knowledge that best reflects student abilities and knowledge. Therefore, much of this thesis is about student modeling. Similarly, Clancey (1986) emphasizes the creation of a model that can simulate student processing and that separates the general or domain model from the inference procedure. Using strategic rules (the inference procedure) and computation rules (the general model), al-Khorezmi can simulate the student processing that we have observed in our data. However, al-Khorezmi does not create a simulated student by creating a rulebase from a subset of the rules in our strategic and computation rule models. Instead, al-Khorezmi observes and records student usage of the rules in our models so that it has the option of choosing to use only rules that it believes the student has used or of applying any rule, since all the rules are in the rulebase at once. Therefore, our purpose here is to explicate the record
keeping done to record which parts of the total knowledge base a particular student holds.

6.2.1. Modelling domain knowledge

Recording the computation rules applied by a student is rather straightforward. When the student gives an expression to be applied to the problem, the expression makes explicit which computation rule the student is applying, and al-Khorezmi records, in the student model the usage of the computation rule. When the student gives a calculated result, recording which rule the student is applying is a little more difficult. The difficulty lies in the possibility of a result being calculated in more than one way in the problem-solving state in which the calculation is done. At present, there is no way to verify which quantities and expression are employed. However, providing a calculator on the screen, for the student to make his calculations on, would enable the system to capture the student's entries into the calculator. Those entries determine which quantities and expression the student employed, thereby enabling determination of which computation rule he applied.

Even if only one rule applies, the record of correctly used rules can never be completely accurate on the give-calculated-result option. Reasons for this are:

- Quantities in some problems may be such that even when they are combined incorrectly a calculated result may be correct.
- Give-calculated-result assumes the student has calculated the result directly when he may in fact have used an indirect method.
- Students sometimes use the give-calculated-result option to give results requiring more than one computation rule.

This last possibility requires a system to guess what computations the student actually performed. In this case, the technique that uses the fewest
computations to compute the quantity is the best approximation to the
computations the student actually performed. This can be implemented with
an algorithm for finding the shortest path of calculations in the graph formed
by the applicability of the computation rules to the known quantities in the
problem.\textsuperscript{35}

Currently al-Khorezmi records all single computation rule applications
that explain the action taken by the student. This gives a nearly accurate
record of the student's application of computation rules. It is not totally
accurate, because sometimes more than one rule is applicable, in which case
we record all such rules, and sometimes a student applies more than one rule,
in which case al-Khorezmi does not determine the sequence of rules that the
student applied.

6.2.2. Modeling strategic knowledge

Recording strategic rule usage is less accurate, because currently rules
from more than one strategy can describe the same action performed by the
student. A state in which a rule and its refinement both apply is a classic
example of a state in which more than one rule applies. This is because the
simpler rule has the same conditions as the refined rule, except the refined
rule has one or more additional, restrictive conditions. For our strategic rule
sets, we only trace the refined rules, because we consider the simplified rules
and the general rule-selection and solving-action rules to be too general to
make tracing practical.

It is also possible that analogous rules from different strategies describe
the action performed by the student. For example, the student's observable

\textsuperscript{35}We have not implemented this capability at this time, as the need for it is rare.
behavior does not distinguish whether he is using strategic rule FD2.1 or strategic rule ME2.1, which are given in Figure 6.5. Before it could distinguish which rule he used, the system would need to ask the student whether he selected the rule because it contained the goal quantity, or whether he selected the rule because it contained known quantities in the problem. In situations where we cannot distinguish whether the student used one or the other of two analogous rules, we record a usage for each and depend on the rules being analogous (and therefore reflecting similar behavior) to keep the record from deviating too far from a good approximation of the student's strategic ability.

FD2.1: Given a set of attributes with known values then select a rule with those attributes as its inputs and determine consistent bindings.

FD3.1: If all input attributes of a bound computation rule have known values then perform the computation rule.

ME2.1: Given a goal attribute then select a rule computing its value as output and determine consistent bindings.

Figure 6.5. Strategic rules FD2.1, FD3.1, and ME2.1.

Another possibility (which we have not pursued) is to further refine the strategic rules and make appropriate changes to the rule sets to remove overlap in the applicability of the rules. We have not pursued this possibility, because we have no evidence that it would be beneficial to teach rules this finely refined. With three levels of strategy rules, our rules already have a higher degree of refinement than those taught in the classroom.

The above discussion applies to determining which rule-selection strategy rule a student applied when he gave an expression. When the student

36Chapter 3 discusses how we distinguish which strategy a student uses.
gives a calculated result, al-Khorezmi makes the assumption that FD2.1 has been applied to select and bind the computation rule and solving-action rule FD3.1 has been applied to take the bound computation rule and perform its computation.

Al-Khorezmi records application of the other solving-action rules when the student chooses the solve option from the menu and requests a particular solving action. The knowledge of when solution techniques can be applied to the expressions that have been created is important and useful to algebraic manipulation in general, not just to solving word problems.

The other two categories of strategic rules, for noting goals and reporting answers, are not as significant. Rules in these categories are few and are not exploited to determine a student’s ability nearly as much as rule-selection and solving-action strategic rules are.

6.2.3. Modeling misconceptions

Not only does al-Khorezmi record the correct usage of correct rules, but it also records the incorrect usage of correct rules, the correct usage of deviant rules (mistaken analogies), and the incorrect usage of deviant rules. Our model of bugs as simplifications of correct rules predicts incorrect applications of correct rules that may occur. Procedures represent incorrect application of correct rules by producing the result of these incorrect applications of correct rules for comparison to the action that the student took. When al-Khorezmi checks for the incorrect usage of a correct rule, it calls the procedures that generate, from the correct rule, the types of errors that we have observed and compares these errors to the student’s work. Because we

37 Chapter 5 discusses the classes of correct and incorrect or deviant rule usage and the uncovering of the origin of errors for providing corrective feedback.
have found that students commit the same kind of errors when applying mistaken analogies as they commit when applying correct rules. Al-Khorezmi can check for errors in both situations in the same manner. Al-Khorezmi makes a record of each type of incorrect rule usage by a student and stores it with the corresponding correct rule. We employ a flavor that mixes in the flavor for correct rules to represent deviant rules. Through inheritance, recording incorrect use of deviant rules employs the same techniques as recording incorrect use of correct rules. Not only are the methods for recording the incorrect usage inherited, but so are the methods for explaining the incorrect usage. For example, the increment method, which is given in Figure 6.6, is applied to update the count of the number of times a rule has been used. Since incrementing an entry in the database is a general operation that is not restricted to recording computation rule use, increment is an operation of flavor db-utility that is mixed into flavor general-rule. Furthermore, flavor general-rule is mixed into flavors comp-rule and strat-rule. An example of a method for explaining incorrect rule usage is the computation rule method mis-bound (also given in Figure 6.6) that is used to describe incorrect bindings for flavor comp-rule or, by inheritance, for flavor deviant-comp-rule.
(defmethod (db-utility increment) ; Increment the quantity stored at
(access-path) ; access-path in the current object.
(>>, (access-next-to-last self access-path))
(car (last access-path))
(addl (access self access-path))) ; The current object is updated and so must
(fact refresh-model ^((access-next-to-last self access-path))) ; be its database fact.

(defflavor comp-rule
  ((description "*UNBOUND*"
    (operator "*UNBOUND*"))
   (general-rule) ; Computation rules are formed from general-rules.
   :gettable-instance-variables
   :settable-instance-variables)

(defflavor deviant-comp-rule
  ((reason-deviant "*UNBOUND*"))
  (comp-rule) ; Deviant computation rules are formed from correct comp-rules.
  :gettable-instance-variables
  :settable-instance-variables)

Figure 6.6. Handling misconceptions.

6.2.4. Using the record of student knowledge and ability

Two help systems exploit the record of a student's knowledge and ability to aid in the selection of the material for providing assistance. Because the first help system waits for the student to select the kind and depth of information that he wants to see, we call it the passive help system. This help system emphasizes extending the student's skills by first offering, for student selection, new knowledge that al-Khorezmi has no record of the student using. Because the second help system takes the initiative and applies the problem-solving model to the current problem-solving situation to determine the hint
that it presents to the student, we call it the active help system. This help system emphasizes practicing the skills that the student has already exhibited by first suggesting techniques that al-Khorezmi has a record of the student using.

**Passive help system**

When a student requests help from the passive help system, al-Khorezmi offers help on strategy or tactics (computation rules) by giving instruction on rules it has not observed the student using or by reviewing rules it has observed the student using. Throughout the system of menus used by the program to communicate with the students, we emphasized the preferred options (those options that should usually be selected first) by placing them first in the menu. Therefore, the menu for the passive help system lists the option for help on new rules first and the option to review rules that al-Khorezmi has observed the student using second. Al-Khorezmi consults the genetic graph to determine the frontier of the student's knowledge so that the order in which it selects new knowledge for instruction does not go beyond the student's level of ability. For example, in Figure 6.7 student DH encounters difficulty and asks for help. Al-Khorezmi offers her a selection, which is headed by selections for information on new topics, of the type of help available and waits for her to choose. Because this is the initial interaction of al-Khorezmi with her, it has no record of her applying any rule. Therefore, at \( \{1\} \) al-Khorezmi starts at the beginning of the genetic graph and with the weak forward-directed strategy.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Which option would you like to have explained? 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10? 1

At various stages in solving a problem there are quantities you want to calculate or solve for. This option should be used to note to the system which quantities you are trying to obtain. You will be prompted for Quantity? to which you should answer with the letter from the menu of quantities.

Option? 1 Quantity? b
OPERATION VERIFIED TO BE CORRECT

Option? 2 Quantity? f Value? 10
That constant belongs to quantity amount-20%-solution.

Option? 2 Quantity? c Value? .2
OPERATION VERIFIED TO BE CORRECT

Option? 9
Would you like help on:

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu

A rule applicable to your request is FD2.1: if THERE IS A RULE WITH KNOWN VALUES FOR ALL ITS INPUTS then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS.

Figure 6.7. Passive help for student DH (repeat of Figure 4.18).
Would you like to see:

1) Specific usages of the rule for this problem (if any)
2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any)
4) An analogous rule (if any)
5) More rules that satisfy your last request (if any)
6) a return to the previous menu

THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE PROPORTION OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0 is a rule that has as input the known entities 1.0 & proportion-alcohol-20%-sol.

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2

THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION is a rule that has as input the known entities proportion-alcohol-20%-sol & amount-20%-solution.

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 1

amount-alcohol-20%-sol( ) = proportion-alcohol-20%-sol( ) * amount-20%-solution( )

Would you like to see values with that equation (y or n)? y

amount-alcohol-20%-sol(Unknown) = proportion-alcohol-20%-sol(0.2) * amount-20%-solution(10)

Would you like to 1) return to the previous menu, 2) see another example of the strategy? 1

Would you like help on:

1) strategies you have not used
2) computable relationships for this domain that you have not used
3) reviewing strategies you have used
4) reviewing computable relationships for this domain that you have used
5) return to the main menu

Option? 4 Quantity? b Value? 2
OPERATION VERIFIED TO BE CORRECT

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason? 1

Would you like to 1) stop now 2) load a new problem? 1

Figure 6.7. (continued).
The active help system gives hints on strategies and tactics to apply in solving the current problem. To give the student a better chance at following through on a hint, the help system consults its record of student rule application to find rules

- that apply to the current problem solving situation,
- that correspond to the level of ability al-Khorezmi believes the student is working at, and
- that the student has used.

Then it applies these rules with the hope that "familiarity will breed invention," that is that the student will realize how to proceed and solve the problem himself. Only if there are no applicable rules that the student has applied will al-Khorezmi give hints based on rules the student has not applied.

In such a situation, al-Khorezmi employs the frontier of the genetic graph to select a rule for presentation. The use of the help system, given in Figure 6.8, demonstrates the active nature of the help system. Immediately after the student selects the help-hints option, al-Khorezmi selects and suggests a rule for him to apply.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

Option? 7

Try to use the strategy rule: if THERE IS A RULE THAT USES AS INPUT A PRIMARY GOAL OR A SUBGOAL then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS; to find a rule that will find amount-pure-solution

Choose give-hints again for a more detailed hint.

Option? 7

The computation rule: THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION; can be used to find amount-pure-solution

Choose give-hints again for a more detailed hint.

Option? 7

The goal amount-pure-solution can be found from:  

\[
\text{amount-alcohol-pure-sol( ) } = \text{proportion-alcohol-pure-sol( ) } \times \text{amount-pure-solution( )}
\]

Choose give-hints again for a more detailed hint.

Option? 7

The goal amount-pure-solution can be found from:  

\[
\text{amount-alcohol-pure-sol(Unknown) } = \text{proportion-alcohol-pure-sol(1) } \times \text{amount-pure-solution(Unknown)}
\]

Figure 6.8. The active help system (excerpted from Figure 3.11).
6.3. Using the frontier in the student model for giving hints

Figure 6.8 shows al-Khorezmi giving hints aimed at helping a student set up expressions. Al-Khorezmi selects the hints that it does, because rules that lie on the frontier of the student's knowledge represent the knowledge used in the actions suggested by al-Khorezmi. The determination that this knowledge lies on the frontier of the student's knowledge comes from observations of the student's problem solving and the genetic graph, which relates the knowledge suggested by al-Khorezmi to the knowledge observed in the student's problem solving. Even though the emphasis of the active help system is to give hints that exploit knowledge the student has applied before, here al-Khorezmi gives hints that employ knowledge on the frontier of the student's knowledge, because there was no combination of applicable strategy and domain rules that the student had applied before.

Table 6.1 gives the record of rule application that al-Khorezmi kept on the person using al-Khorezmi, which led to the help given in Figure 6.8.\textsuperscript{38} The table shows that the user has solved two previous problems, because al-Khorezmi has a record of two applications of FD1 to report that a problem was solved. Figure 6.8 shows that al-Khorezmi determined that the user is able to work at the means-ends level, because al-Khorezmi suggests goal-directed behavior by hinting towards the application of strategic rule ME2.2. Had it not assumed the user was capable of applying the means-ends strategy, it could have suggested to the user several applications of strategic rule FD2.1 in combination with previously applied computation rules. Suggesting

\textsuperscript{38}The system does not completely record usage of ME2.1 and ME2.2 at this time because it requires the student to indicate his subgoals before it recognizes selection of rules to satisfy subgoals as usages of ME2.1 and ME2.2. There are two ways to correct this: 1) force the student to designate subgoals, and 2) implicitly assume nongoal unknowns in expressions containing the goal are subgoals.
application of strategic rule FD2.1 would not attempt to get the user to use the frontier of his knowledge, but would have followed the strategy of hinting at previously applied knowledge.

<table>
<thead>
<tr>
<th>Computation rule</th>
<th>Times used</th>
<th>Strategic rule</th>
<th>Times used</th>
</tr>
</thead>
<tbody>
<tr>
<td>CR1</td>
<td>1</td>
<td>FD1</td>
<td>2</td>
</tr>
<tr>
<td>CR3</td>
<td>3</td>
<td>FD2.1</td>
<td>5</td>
</tr>
<tr>
<td>CR5</td>
<td>1</td>
<td>FD3.1</td>
<td>7</td>
</tr>
<tr>
<td>CR8</td>
<td>1</td>
<td>ME2.1</td>
<td>1</td>
</tr>
<tr>
<td>CR10</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6.1. Modeled rule usage for user in Figure 3.11.

Al-Khorezmi hints at and then presents the rules it does, because there is no combination of a means-ends strategy rule and a computation rule that has been previously applied by the user. Specifically, since the goal is to find the amount of pure alcohol, the candidate base rules are CR1, CR3, and CR10, (these computation rules are repeated in Figure 6.9) but none of these have the amount of pure alcohol as their output. Therefore, al-Khorezmi has the option of presenting the unused strategy rule ME2.2 (given with ME2.1 in Figure 6.10) with the used computation rules CR1, CR3, and CR10 or of presenting the used strategy rule ME2.1 with the unused computation rules CR2, CR4, and CR11. We made no distinction in the programming of these options. Therefore, the conflict resolution of the production system monitor made the choices seen in Figure 6.8. Any redundant, equivalent choices are not given as hints, because we programmed al-Khorezmi to delete them. If there were a combination of an applicable means-ends strategic rule and a computation rule, each of which the user had used, then al-Khorezmi would choose those rules for hints before it would choose an unused rule.
CR1: The amount of a solution is equal to the sum of the amounts of its parts.

CR2: The amount of one part of a solution is equal to the amount of the solution minus the amount of the other part.

CR3: The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.

CR4: The amount of a solution can be found by dividing the amount of the substance in the solution by the proportion of the substance in the solution.

CR5: The proportion of a substance in a solution can be found by dividing the amount of the substance in the solution by the amount of the solution.

CR6: The sum of the proportions of the substances in a solution is 1.0.

CR7: The proportion of a substance in a solution is 1.0 minus the proportion of the other substance in the solution.

CR8: The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating solutions.

CR9: The amount of a substance in a solution is the amount of that substance in the composite solution minus the amount of that substance in the other solution in the mixture.

CR10: The amount of a solution is the sum of the amounts of the two participating substances.

CR11: When two substances are mixed to form a solution, the amount of one of the substances is found by subtracting the amount of the other substance from the amount of the solution.

Figure 6.9. Computation rules for wet-mixture problems (repeat of Figure 5.1).

ME2.1: Given a goal attribute
       then select a rule computing its value as output
       and determine consistent bindings.

ME2.2: Given a goal attribute
       then select a rule having it as input
       and determine consistent bindings.

Figure 6.10. Strategic rules ME2.1 and ME2.2.
Strategic rule ME2.2, which al-Khorezmi chose to present, is at the frontier of the student's knowledge because it is an unused strategic rule at his current strategy level. Computation rules CR2, CR4, and CR11, which al-Khorezmi could have chosen to present, are at the frontier of the user's knowledge because they are one derivation link from computation rules CR1, CR3, and CR10, which the student has applied.
7. Conclusion

In conclusion we review the goals that we established in the introduction and evaluate the results that we obtained while working on those goals. We also evaluate al-Khorezmi by comparing it to the criteria that Clancey (1986a) proposed for intelligent tutoring systems. We conclude that al-Khorezmi meets these criteria. Finally, we point out some of the shortcomings of al-Khorezmi and of the research that need additional work, and we propose directions for further investigation of knowledge organization for intelligent tutoring and for further development of al-Khorezmi.

7.1. Review of the goals

The primary goal of this research is to investigate an appropriate organization for building an intelligent tutoring system. Our particular emphasis is on setting up computational relationships that are employed to solve algebra word problems. Two separate areas that we investigated in pursuit of our goal are

1) pedagogy, and

2) knowledge representation, content, and organization.

Of these we gave most of our attention to knowledge content and organization.

Because private tutors are the most effective method for providing instruction (Bloom, 1984), our pedagogical subgoal is to attain a similar level of effectiveness with an artificially intelligent tutor by providing
• an environment in which a student can practice solving problems and
• corrective feedback at the student's level of knowledge when the student
  errs.

Our subgoal of uncovering the proper knowledge content and
organization to provide the most effective tutor possible for a procedural task
led to investigation of four areas. These areas are:

1) the separation of strategic and tactical knowledge in a problem-
solving model,

2) the progression of knowledge and ability from novice to expert
   in a problem-solving model,

3) the genetic relationship of new knowledge to previously
   acquired knowledge, and

4) the determination and explanation of student errors.

In the next section, we evaluate the results that we obtained while working on
these goals.

7.2. Evaluation of the results

Our research on algebra word problem solving led to the development of
a rule-based, problem-solving model. This model is important, because it (as
opposed to less formally specified models) provides educators with a well-
defined statement of how students solve problems. Furthermore, this
statement details the separation of the strategy from the tactics and delineates
the progression of ability from novice to expert. Therefore, the amount of
explicit detail that this model provides enables a better understanding of this
task. In addition to use in an intelligent tutor, such a model can be exploited
to better evaluate student progress by identifying a qualitative measure given by the level of strategy students are capable of as opposed to a quantitative measure of the number of problems of a particular type that have been solved correctly,

to better understand their mistakes by explicating the problem-solving process to the point that it can be determined what component of the process that the mistake is a variant of,

to better understand the relationship between the instructional materials employed and what the students learned by determining what components of the model that the instructional materials employ have been learned by the student,

to design instructional materials that support obtaining primary instructional goals by identifying the component steps of the procedure that achieve the effect the goals identify, and

to help select the important components of the model to be taught, because the model helps identify characteristics such as generality, transferability, and difficulty.

Our having devised a technique for determining which strategy a student employs has implications beyond just identifying the level of the student's processing. In situations where multiple strategies are possible and predictions of future behavior are to be made, having identified the strategy being employed can be exploited to exclude the necessity of making predictions from the other strategies.

Determination that a student's improvement from novice skills to expert skills is not done by rule refinement as Goldstein (1979) proposed led to revising the genetic graph by replacing refinement as the learning and organizing link with composition, combination, and sophistication. Anderson (1986) has demonstrated the role of composition in learning through knowledge compilation. Combination is a variant of composition not covered by Anderson's definition. Sophistication is a new relationship that works in conjunction with the new definitions of analogy to define a transformation in knowledge that adds capabilities instead of just compiling them.
Brown and Burton (1978) have shown that students exhibit interacting bugs. Our approach supplies a way to produce interacting bugs without having to form all pairs of bugs as Brown and Burton (1978) did. To do this we define one class of simple bugs as those bugs generated by incorrect application of correct rules. Certain other rules that students hold are incorrect. They believe these rules to be correct, because there is a strong correlation of these incorrect rules to correct rules. When these incorrect rules are applied correctly, they form the second simple bug class. By applying the incorrect rules incorrectly, we are combining one class of bugs with another thereby generating the interacting bugs that we have observed. The repair theory of Brown and VanLehn (1980) does not generate any of the interacting bugs of Brown and Burton (1978).

The success of our uncovering, modeling, and providing feedback on mistakes fulfills the pedagogical subgoal of providing corrective feedback. Furthermore, al-Khorezmi creates a practice field on which students can practice problem solving, thereby fulfilling the other pedagogical subgoal. This practice field is not ideal, because it does not allow the student to input the problems he wants to solve. This capability can eventually be provided by integrating al-Khorezmi with one such as Algebra Reader (Rapp, 1986) that creates an appropriate problem representation.

7.3. Meeting Clancey's criteria

Our goal here is to demonstrate the extent to which we meet the criteria set forth by Clancey (1986) in a seminal paper on general concepts applicable to the construction of intelligent computer-assisted-instruction systems.

* This section sets our work in the context of Clancey (1986a) and may be skipped without losing continuity.
Clancey employs the concept of qualitative student models to define a variety of models to base intelligent computer-assisted-instruction systems on and to compare the strengths of such systems. Al-Khorezmi employs a qualitative student model that is based on the problem-solving procedure that the students carry out. This is as opposed to a quantitative model that records, for example, the number of correct and incorrect answers. According to Clancey,

A qualitative student model describes a student's knowledge structurally, in terms of relations among concepts and a problem solving procedure (1986, p. 381).

Clancey proposes that the term "qualitative model" has been too restrictively applied to simulation models of physical processes (Bobrow, 1984), and that it is possible to distinguish other types of qualitative models besides simulation models of physical processes. An example is the classification model of the MYCIN system (Shortliffe, 1976) and of other systems described by Clancey (1985). Clancey also points out that there are other qualitative simulation models than those for physical processes.

Because al-Khorezmi can carry out the same problem-solving steps that the students perform, it provides a simulation model of student problem solving. Therefore, it can

- employ its model of the student to select hints, by employing the active help system, that the student can follow through on,
- explain student behavior by following student performance and matching that behavior to the rules it employs for modeling, and
- exploit its ability to follow the student to make it easier to uncover and explain errors when they occur.

These abilities correspond to what Clancey points out as the importance of simulation models of reasoning for instructional systems:
they can be exploited to predict behavior,
they can explain the basis of observed behavior, and
inferring errors from a simulation of reasoning is more efficient than pre-enumerating them in a classification.

We have divided the rules of our model into computation rules and strategic rules. This division has proven valuable, because:

- it is more important for the student to learn the problem-solving strategies, which being separate are more accessible for tutoring, independent from the tactics than to learn problem-specific tactics or formulations with strategies compiled into them,
- a separate strategic component enables distinguishing separate strategies that differ in power, thereby allowing representation of novice through expert performance, and
- general strategic rules enable implementation of computation rules specific to each of several word problem domains.

For instructional systems, Clancey advocates simulation models of reasoning constructed from a general model and an inference procedure. Clancey (1986) states, "The general model describes what is known about the world." In al-Khorezmi the general model is the set of computation rules, which describe computational relationships that hold between various types of quantities. The inference procedure "focuses and orders the gathering of problem information and making assertions about the solutions" (Clancey, 1986). The strategic rules are the inference procedure of al-Khorezmi. The application of the inference procedure to the general model on a specific problem produces the situation-specific model, which in al-Khorezmi is the calculations and equations that are employed in solving a specific problem. According to Clancey the separation of these three components, which we have achieved in al-Khorezmi, is an ideal because:
it is obvious where situation-specific assertions originate,
the program can articulate both the strategy and the general model,
the reasons for believing the general model (in our case it is axiomatic)
can be represented, and
information such as efficiency can be represented for the inference procedure.

Clancey continues by saying that a complicating factor for instructional programs dealing with human learning is the ability of students to become more efficient through practice, instead of just learning new facts. He concludes that systems based on expert reasoning alone are inadequate and the ability to model student processing different than the ideal expert model is "crucial." The different strategies of our strategic model provide this "crucial" capability.

Al-Khorezmi emphasizes providing helpful corrective feedback. When a student errs, not only do we point out the student's error, but we state why the operation is in error and how it is possible for the student to incorrectly formulate such an erroneous operation. As an example, consider the response of al-Khorezmi (given in Figure 7.1) when it encounters a mistake made by a student during the experimental use of al-Khorezmi. Al-Khorezmi first points out the discrepant behavior—that the equation the student employed was invalid {1}. It then points out the bug in the student's problem solving knowledge—thinking that it is possible to sum, from different solutions in a mixture, the proportion of a substance in those solutions {2}. Next, al-Khorezmi gives an explanation of how the bug might have come about by suggesting that the student made a mistaken analogy with the legal operation—summing, from different solutions in a mixture, the amounts of a substance in the solutions {3}. Finally sometimes, such as in this example, al-Khorezmi gives instruction on self-monitoring techniques {4} by suggesting that the
student can reason about mixing solutions and realize that there will be an averaging effect on the proportions.

\{3\} The following equation was not found to be valid:
\[(0.2 + x_1) = 0.7\]
\{2\} You used the following ILLEGAL rule: WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE PROPORTION OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING THE PROPORTIONS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS to combine 0.2 of proportion-alcohol-20%-sol with \(x_1\) of proportion-alcohol-pure-sol. \{3\} This rule is wrong because: Summing the amounts of these substances is legal (cr8) but it is a mistake to make an analogy and sum the proportions of these substances. \{4\} You should monitor your use of computational relationships. When two solutions are mixed, the proportion of a substance in the solution falls between the proportions of the substance in each of the solutions.

---

Figure 7.1. Al-Khorezmi's response to student mistake (from Figure 5.6).

Clancey delineates three levels of description for diagnosing discrepancies, with diagnosis playing a "central" role in teaching. These three levels are:

1) the discrepant behavior that appears in the student's work,
2) the bug in the student's processing, and
3) the explanation of how the bug came about.

As seen above, al-Khorezmi describes discrepant behavior on these three levels, but goes one step further by providing an explanation of how to avoid the bug in the future.

We found that simplification of the conditions of a rule is a plausible explanation for the errors in student performance on word problems.\(^{39}\) Mistaken analogy is a more extreme form of simplification that also explains errors in student performance. These results provide a basis for a generative theory of bugs. Al-Khorezmi exploits this theory to generate bugs when a

\(^{39}\)Chapter 5 presents an analysis of errors in student problem solving that shows a plausible explanation of student errors is rule simplification.
student makes an error. If one of the bugs matches the student's error, then al-Khorezmi has an explanation for the error. This theory has not been completely developed, nor rigorously tested, but it generates rules matching most of the discrepant behavior found in our data and matching all of the discrepant behavior found in experimental use of al-Khorezmi. Therefore, our theory satisfies the present demands on it by giving a plausible explanation of the origin of a student's discrepant behavior. The importance of the development of such a generative theory of bugs is stated by Clancey, who says that a generative theory of bugs would solve the problems of:

- missing bugs in a bug catalog or pre-enumerated set of bugs, and
- simply too many bugs to pre-enumerate.

The prevalent model for bug origin is that reasoning is not random, but that there is a causal explanation for every error. The three categories of explanations for bug origins are mis-learning, construction, and slip. Mis-learning is caused by a learning error, whether it is from an over-generalization (simplification) or a false or mistaken analogy (Matz, 1981; Sleeman, 1984). Construction is caused by weak or incomplete methods leading to an impasse and having a repair take place (Brown and VanLehn, 1980). A slip may be caused by fatigue, "cognitive overload," or other such causes that make a student err when he knows better (Clancey, 1986a). Our observation of student problem solving found students who reached an impasse caused by weak methods (the weak forward-directed strategy) and made a repair employing simplification, which is attributed to mis-learning. Therefore, we argue that construction of a repair and mis-learning are closely related and anticipate that generating simplifications will be sufficient to find explanations for the errors our students make.
Al-Khorezmi assesses a student's work in several ways. Optimally a system should provide as complete an assessment of a student's work as possible. To determine how complete an assessment al-Khorezmi makes of a student's work, we compare it to the logical spectrum of assessments of a student that instructional programs can infer that Clancey (1986a) gives, which are repeated in Figure 7.2. This comparison shows that al-Khorezmi covers each of the six categories of assessments given by Clancey in some way, thus demonstrating how completely it assesses a student's work.

<table>
<thead>
<tr>
<th>1. Determine whether statements about the problem to be solved and the general model are correct.</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Determine whether a student's statements about the problem are correct.</td>
</tr>
<tr>
<td>b. Determine whether a student's statements about the general model are correct.</td>
</tr>
<tr>
<td>2. Determine whether a student's solution (situation-specific model) is consistent with the general model and the situation-specific information he has received.</td>
</tr>
<tr>
<td>3. Determine what knowledge (domain model and inference procedure) the student has used to form a partial solution (make a hypothesis).</td>
</tr>
<tr>
<td>4. Determine which incorrect general model is consistent with the student's behavior.</td>
</tr>
<tr>
<td>5. Determine what nonoptimal or incorrect inference procedure is consistent with the program's general model and the student's behavior.</td>
</tr>
<tr>
<td>6. Determine what underlying misconceptions explain the student's incorrect general model or buggy inference procedure.</td>
</tr>
</tbody>
</table>

Figure 7.2. Clancey's spectrum of student assessments.

Al-Khorezmi determines whether the student makes correct statements about the problem to be solved (assessment 1) by determining whether the student has noted the constants and goals of the problem correctly. In our present interface the student makes no statements about the general model and, therefore, al-Khorezmi makes no evaluation at that level. The interface that we developed does keep all situation-specific information available for display, and as the user enters each new piece of situation-specific information, al-Khorezmi evaluates it for consistency with the general model.
and the current situation-specific information (assessment 2). To determine what knowledge the student has employed to form a partial solution (assessment 3), we exploit model tracing to record all domain model (computation rules) and inference procedure (strategic rules) usage. Because of the formal nature of our domain, it is possible to determine when the student makes a mistake. Furthermore, in almost all cases, we can determine, by applying the results of our analysis of student problem-solving bugs, which incorrect general model is consistent with the student's behavior (assessment 4). Our analysis of problem-solving data has not led to any concrete conclusions about incorrect inference procedure usage, but we have implemented two nonoptimal levels of inference procedure (strategic rules), and can determine whether a student is employing the optimal level or one of the nonoptimal levels (assessment 5). Finally, al-Khorezmi makes use of our hypothesis that simplification (mis-generalization in Sleeman's (1984) terminology) and mistaken analogy are the origins of the bugs seen in the incorrect general model or buggy inference procedure (assessment 6).

Because the general model is axiomatic and the inference procedure is algorithmic in the domain of algebra word problems, it is possible to develop a more complete model than it is possible to develop for domains such as programming and medical diagnosis. Clancey contrasts formal domains, such as algebra word problems, with domains which do not have an axiomatic general model, such as chemistry and physics theory; with domains that have an axiomatic general model, but a heuristic rather than algorithmic inference procedure, such as the use of MACSYMA (Mathlab, 1977); and with domains that have neither an axiomatic general model, nor an algorithmic inference procedure, such as medical diagnosis. He concludes that we must temper our success with mathematical modeling with the realization that less formal
domains have bugs that are more difficult to ferret out and that results in modeling heuristic inference procedures are tentative and incomplete.

Al-Khorezmi separates the inference procedure from the general model. This is not necessary to model problem solving which employs different strategies, because the strategic component could be encoded into the computation rules to produce identical surface behavior. Separating the strategies out makes our model a functional simulation model as opposed to a behavioral simulation model (Clancey, 1986a). From this separation we obtain the advantages of functional models over behavioral models. According to Clancey, these are:

- the general model can be explained separately from the inference procedure,
- the ability to selectively interpret the general model, thereby decreasing the combinations of situations that must be anticipated, makes the system more robust, and
- the student model can be constructed with a separation of inference procedure and general model, thereby reducing the number of combinations of situations where gaps or misconceptions may occur in the general model.

In this section we have shown that the accomplishments of our work match the optimal criteria set forth in Clancey (1986a). We now turn our attention to the limitations of al-Khorezmi and the attendant directions for further research and development.

7.4. Limitations of al-Khorezmi

Because the problem-solving model lacks the information and techniques that expert problem solvers employ to monitor their processing, the instructional capabilities of al-Khorezmi are limited. If the goal is to improve student problem-solving abilities, then all of the techniques that experts bring to bear on problem solving should be taught. Self-monitoring is
an important technique that experts employ, but which students do not exploit. Without self-monitoring abilities, students believe they have solved problems correctly when they have not. Also, they cannot see why their teacher's correct solution is better, because they cannot find fault with their own solution (Lester, 1985).

Al-Khorezmi currently is a practice field with only two ways in which the student can advance his learning. These ways are:

1) the student takes the initiative and explores advanced strategies on his own by employing the passive help system, or

2) al-Khorezmi presents a difficult problem to the student. If the problem is beyond the student's level of processing, then when the student requests help from the active help system, al-Khorezmi must present more sophisticated techniques than the student currently knows.

Currently al-Khorezmi has no capabilities for instructing new techniques beyond these two. In Appendix D we outline a proposal for the design of an extension, to make al-Khorezmi into a teaching system, that would extend the capabilities of al-Khorezmi to the instruction of new techniques and knowledge.

Although al-Khorezmi maintains a quantitative student model that records the student's employment of the computation and strategic rules and the types of errors the student makes, the information in this student model is not fully exploited because:

---

40 The system is further limited because its implementation of the active help system is only extended into the means-ends strategy and not on to the expert strategy.
• there is no general teaching facility at present (the instruction that the student model supports is currently limited to the two help systems) and

• al-Khorezmi currently generates corrective feedback based solely on the immediate error without consulting this student model for the student's history of processing.

Although al-Khorezmi contains genetic links, these links have limited functionality because they are hard-coded with canned descriptions of the relationships between the rules connected by the links. With a more generalized approach that represents the genetic graph relationships procedurally, the links could be established dynamically when needed, and the descriptions could be generated dynamically to state the specifics of the situation as opposed to employing canned descriptions couched in general terms.

The user interface is based on a menu system and a linear input/output capability. Because we put the primary research and development effort into other facets of al-Khorezmi, such as the problem-solving model, the genetic relationships, and the diagnostic capabilities, we did not take the time to create a sophisticated interface. Therefore, the interface is far from ideal. However, when the students from Lutheran High School put al-Khorezmi to experimental use, those who had experience employing other menu-driven programs caught on to employing the menus of al-Khorezmi very quickly.

A final limitation of the current version of al-Khorezmi is the slow speed at which it operates, especially when verifying expressions or equations entered by a student. We believe that the primary reason for inefficiency in this situation is the dependency on rule-based programming to implement all of al-Khorezmi, whereas parts such as expression verification could be done faster with an algorithmic approach.
7.5. Directions for further research

Further investigation must be done to determine and implement tactics and strategies for self-correction and monitoring skills. Experienced problem solvers consistently apply these skills to determine:

- when they have made a mistake through simplification or analogy,
- when they have achieved a correct solution, and
- when they have reached a dead end and need to backtrack and try a different tactic or strategy.

Therefore, if we want our students to obtain expert skills, we must instruct them in this very important facet of expert problem solving.

If adding self-monitoring skills to the instructional capabilities of the tutor will lead to determining the need to try a different strategy, then the tutor must have other strategies that it knows and can tutor, as opposed to only variations on the same algebraic formulation strategy that is currently implemented. One such strategy is the hill-climbing strategy, which provides a way to solve problems without explicitly creating formulas and algebraically manipulating them. For example, the problem

How much solution is there in a 36% alcohol solution containing 10.8 gallons of alcohol?

can be solved by estimating 50 gallons, checking with the computation

\[ .36 \times 50 = 18, \]

estimating 40 gallons, checking with the computation

\[ .36 \times 40 = 14.4, \]

estimating 30 gallons, and checking with the computation

\[ .36 \times 40 = 10.8. \]

\[ ^{41} \text{Chapter 3 presents a version of the hill-climbing strategy that deals with problems with one unknown and discusses how the strategy could be extended to handle problems in two unknowns.} \]
To implement such a strategy in the tutor, we would add to the menu of options an estimate-goal-and-check option which would ask the student:

1) estimate which quantity?

2) as what value?

3) check against which quantity?

Given two quantities, the system could determine the intended computation, carry out the computation, and provide the value computed for the student to check against the known quantity. The student could repeatedly apply this option until the proper value results from the estimation.

For this hill-climbing option, errors could be noted to the student that stem from:

- there being no existing computation rule that allows checking against the quantity the student specified,

- making the next guess in the wrong direction after receiving the computed check value, and

- more than a few estimates being required in which case it could be noted to the student that perhaps he should now apply algebraic techniques, because hill-climbing was proving to be more difficult and the more powerful techniques would be more suitable.

Chapter 3 presented a formal statement of the hill-climbing strategy for problems with a single unknown, the unknown being estimated. The implementation of this strategy will exploit techniques for selecting rules that are employed by the weak forward-directed strategy. We anticipate that implementing a formalization of the hill-climbing strategy for problems with two unknowns will similarly exploit techniques developed for the means-ends strategy implementation.

---

42This has been shown possible because the system already determines the intended computation rule for the give-calculated-result option from the single resultant quantity.
Other guess-and-check methods, such as systematic consideration of all possibilities and inductive examination of all cases (Butts, 1985), are candidates for providing optional strategies to exploit in solving problems. To implement these, we need to collect data while students solve problems for which these strategies are appropriate, and we need to identify the characteristics that determine when a particular strategy is appropriate.

Another obvious direction for further research is to progress from the current model-tracing tutor to a full scale teaching system that implements a curriculum encompassing all variations on the algebraic formulation strategy and eventually other strategies as well. In Appendix D we present our ideas on how to dynamically derive a curriculum from a rule-based model of problem solving by making use of the felicity conditions of VanLehn (1987) and genetic relationships between rules (Goldstein, 1979) in the model. Implementing this dynamic derivation requires the development of meta rules that will establish felicity conditions by examining a rule set and determining which rules are applied in disjunction and which rules are applied in conjunction. It also requires the development of meta rules which that will establish genetic relationships by examining a rule set and determining refinements, analogies, and combinations. Because such meta rules build the curriculum dynamically into a structure, the AND-OR graph, on which the rest of the tutoring or teaching can operate, they could lead to the development of practical authoring systems for intelligent tutoring or teaching systems.

Such a system would require generalizing across procedural domains the implementation of our pedagogical principles for giving corrective feedback, adjusting the speed of progress through the curriculum, and explaining new knowledge based on genetic relationships to known knowledge. There would be rules concerning increasing or decreasing the
speed of advancement based on the ability of the student to absorb new material, composing problems from component skills when the student has consistent trouble with a composite type of problem, forming examples by employing the specialization relationship (a capability which is already done dynamically but needs to be generalized), and making explanation of new knowledge by relating it to known knowledge. Other pedagogical principles are inherent in the structure of a tutor based on models that separate the inference procedure from the general model.

Before finalizing a completed design of al-Khorezmi, an investigation into a suitable interface should take place. Because of the amount of time consumed by typing, we believe a natural language interface based on typing input is undesirable. Until it is possible for computers to do (as human tutors do) unlimited voice communication to understand what a student says, and either vision to understand what a student writes, or pattern recognition to recognize what is written on a graphics tablet, we must employ some other interface besides natural language. Given these limitations with respect to interfaces employing human capabilities, investigation into an interface based on graphics and windows appears to be the best choice.

Finally, further investigation must be done on efficiency considerations. This includes consideration of whether each component of al-Khorezmi is more efficient when done by algorithm or rule set, and whether conversion to a more suitable implementation vehicle than YAPS (Allen, 1983) is necessary.43 The continued development of multiparadigm programming

---

43 Appendix C details characteristics of YAPS that led to inefficient execution of the program and our solution to this problem, which sped up the execution, but led to inefficient use of space.
environments and parallel programming systems offers attractive alternative implementation vehicles and possibilities for faster execution.
Bibliography


APPENDICES
Appendix A Computation Rules

CR1: The amount of a solution is equal to the sum of the amounts of its parts.
   Structure: Composite solution C from solutions S1 and S2
   Constraints: S1 not equal S2
   Input values: amount(S1), amount(S2)
   Output value: amount(C)
   Computation: amount(S1) + amount(S2)

CR2: The amount of one part of a solution is equal to the amount of the solution minus the amount of the other part.
   Structure: Composite solution C from solutions S1 and S2
   Constraints: S1 not equal S2
   Input values: amount(C), amount(S1)
   Output value: amount(S2)
   Computation: amount(C) - amount(S1)

CR3: The amount of a substance in a solution can be found by multiplying the proportion of the substance in the solution times the amount of the solution.
   Structure: Solution S with substance P
   Constraints: none
   Input values: amount(S), proportion(P)
   Output value: amount(P)
   Computation: proportion(P) * amount(S)

CR4: The amount of a solution can be found by dividing the amount of the substance in the solution by the proportion of the substance in the solution.
   Structure: Solution S with substance P
   Constraints: none
   Input values: amount(P), proportion(P)
   Output value: amount(S)
   Computation: amount(P) + proportion(P)

CR5: The proportion of a substance in a solution can be found by dividing the amount of the substance in the solution by the amount of the solution.
   Structure: Solution S with substance P
   Constraints: none
   Input values: amount(S), proportion(P)
   Output value: amount(P)
   Computation: proportion(P) * amount(S)

CR6: The sum of the proportions of the substances in a solution is 1.0.
   Structure: Solution S with substances P1 and P2
   Constraints: P1 not equal P2
   Input values: proportion(P1), proportion(P2)
   Output value: 1.0
   Computation: proportion(P1) + proportion(P2)
CR7: The proportion of a substance in a solution is 1.0 minus the proportion of the other substance in the solution.

Structure: Solution S with substances P1 and P2
Constraints: P1 not equal P2
Input values: proportion(P1)
Output value: proportion(P2)
Computation: 1.0 - proportion(P1)

CR8: The amount of a substance in the composite solution is equal to the sum of the amounts of like substances in the participating solutions.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively
Constraints: C2 not equal C3
Input values: amount(S2), amount(S3)
Output value: amount(S1)
Computation: amount(S2) + amount(S3)

CR9: The amount of a substance in a solution is the amount of that substance in the composite solution minus the amount of that substance in the other solution in the mixture.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively
Constraints: C2 not equal C3
Input values: amount(S1), amount(S2)
Output value: amount(S3)
Computation: amount(S1) - amount(S2)

CR10: The amount of a solution is the sum of the amounts of the two participating substances.

Structure: Solution S with substances P1 and P2
Constraints: P1 not equal P2
Input values: amount(P1), amount(P2)
Output value: amount(S)
Computation: amount(P1) + amount(P2)

CR11: When two substances are mixed to form a solution, the amount of one of the substances is found by subtracting the amount of the other substance from the amount of the solution.

Structure: Solution S with substances P1 and P2
Constraints: P1 not equal P2
Input values: amount(S), amount(P1)
Output value: amount(P2)
Computation: amount(S) - amount(P1)

DCR1: The proportion of a substance in the composite solution is equal to the sum of the proportions of like substances in the participating solutions.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively
Constraints: C2 not equal C3
Input values: proportion(S2), proportion(S3)
Output value: proportion(S1)
Computation: proportion(S2) + proportion(S3)
**DCR2**: The proportion of a substance in a participating solution is equal to the difference between the proportion of the like substance in the composite solution and the proportion of the like substance in the other participating solution.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively

Constraints: C2 not equal C3

Input values: proportion(S1), proportion(S3)

Output value: proportion(S2)

Computation: proportion(S1) - proportion(S3)

**DCR3**: The proportion of a substance in the composite solution is equal to the average of the proportions of like substances in the participating solutions.

Structure: Composite solution C1 from constituent solutions C2 and C3 with like substances S1, S2, and S3 respectively

Constraints: C2 not equal C3

Input values: proportion(S2), proportion(S3)

Output value: proportion(S1)

Computation: (proportion(S2) + proportion(S3)) / 2

**Times**: The arithmetic operator times is employed when things are repeated, taken a proportion of, or done at a rate.

Structure: none

Constraints: none

Input Values: Quantity1, Quantity2

Output Value: Quantity3

Computation: Quantity1 * Quantity2

**Simplified-proportion**: When you have a proportion that applies to a certain whole, multiplying the proportion times the whole amount gives an amount of a part of the whole.

Structure: Whole quantity Q which may have part P

Constraints: none

Input values: amount(Q), proportion(P)

Output value: amount(P)

Computation: proportion(P) * amount(Q)

**Simplified-rate**: A rate (other than interest) times a time gives the quantity that was measured in units per time.

Structure: Quantity Q and associated rate and time.

Constraints: numerator(units(rate(Q))) equals units(Q)

Input Values: rate(Q), time(Q)

Output Value: Quantity(Q)

Computation: rate(Q) * time(Q)
Appendix B Experimental Usage

B.1 Student AL

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

a) amount-20%-solution
alcohol b) amount c) proportion d) percent
water e) amount f) proportion g) percent
v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 4 Quantity? b Value? 2

OPERATION VERIFIED TO BE CORRECT
Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.
Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

a) amount-20%-solution 10.0
alcohol b) amount 2 c) proportion 0.2 d) percent
water e) amount f) proportion g) percent
v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason? 1
Would you like to 1) stop now 2) load a new problem? 2
[load /usr1/bayley/tutor/solver/problems/mix-problem5.1]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

a) amount-30%-solution
acid b) amount c) proportion d) percent
water e) amount f) proportion g) percent
i) amount h) amount-15%-solution
water j) proportion k) percent
l) amount m) proportion n) percent
p) amount o) amount-mixture
acid q) proportion r) percent
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

a) amount-30%-solution 10.0
b) amount 3
c) proportion 0.3
d) percent
e) amount
f) proportion
g) percent
h) amount-15%-solution
i) amount
j) proportion
k) percent
l) amount
m) proportion
n) percent
o) amount-mixture
p) amount
q) proportion
r) percent
s) amount
t) proportion
u) percent
v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{Acid} = 10.0 \\
\text{Water} = 20.0 \\
\text{Amount} = 6 \\
\text{Proportion} = 0.15 \\
\text{Percent} = 58\% \\
\]

You used a simplified version of the rule:
THE PROPORTION OF A SUBSTANCE IN A SOLUTION IS THE AMOUNT OF THE SUBSTANCE IN THE SOLUTION DIVIDED BY THE AMOUNT OF THE SOLUTION

You must refine your selection of quantities to amount-mixture instead of proportion-alcohol-30%-solGarbage collecting . . [list:3909{70%}; fixnum:11{8%}; ut:58%]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

- alcohol
- water
- amount-20%-solution
- proportion 0.2
- percent
- amount
- proportion
- percent
- amount-pure-solution
- proportion
- percent
- amount-70%-solution
- proportion
- percent
- amount
- proportion
- percent
- full proportion constant 1.0
- full percent constant 100.0
- other value

A rule applicable to your request is:

\[ \text{if THERE IS A PRIMARY GOAL THAT HAS A VALUE} \]
\[ \text{then NOTICE THAT THE PRIMARY GOAL HAS BEEN FOUND and UTILIZE THE ANSWER TO REPORT RESULT} \]

Would you like to see:

1) Specific usages of the rule for this problem (if any)  
2) A more general statement of the rule (if any)  
3) A more specific statement of the rule (if any)  
4) An analogous rule (if any)
5) More rules that satisfy your last request (if any) 6) a return to the previous menu

5

A rule applicable to your request is: fd3

if THERE IS A KNOWN VALUE AND A RULE FOR WHICH IT IS AN INPUT
then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR UNKNOWNS

Would you like to see:
1) Specific usages of the rule for this problem (if any) 2) A more general statement of the rule (if any)
3) A more specific statement of the rule (if any) 4) An analogous rule (if any)
5) More rules that satisfy your last request (if any) 6) a return to the previous menu

1

THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE PROPORTION OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0

is a rule that has as an input the known entity 1.0

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before?

B.2 Student BM

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

amount-20%-solution 10.0 alcohol

amount 2 water

proportion 0.2

d) percent

e) amount

f) proportion

v) full proportion constant 1.0

w) full percent constant 100.0

y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result

5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 4

Quantity? b

Value? 2

OPERATION VERIFIED TO BE CORRECT

Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.

Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

amount-20%-solution 10.0 alcohol

amount 2 water

proportion 0.2

d) percent

e) amount

f) proportion

v) full proportion constant 1.0

w) full percent constant 100.0

y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result

5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason? 1
Would you like to 1) stop now  2) load a new problem ? 2
[load  /usr1/bayley/tutor/solver/problems/mix-problem2.1]

If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

a) amount-in-solution
alcohol b) amount  c) proportion  d) percent
water e) amount  f) proportion  g) percent
  v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem
Option? 4
Quantity?  b
Value?  5
OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

a) amount-in-solution
alcohol b) amount  c) proportion  d) percent
water e) amount  f) proportion  g) percent
  v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem
Option? 4
Quantity?  a
Value?  15
OPERATION VERIFIED TO BE CORRECT
Installing problem constant 10.0 in amount-water-in-sol as you made use of it in computing your result.

If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

a) amount-in-solution 15
alcohol b) amount  c) proportion  d) percent
water e) amount 10.0 f) proportion  g) percent
  v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem
Option? 10
Do you want to exit because 1) the problem is solved  2) some other reason ? 1
Would you like to 1) stop now  2) load a new problem ? 2
[load  /usr1/bayley/tutor/solver/problems/mix-problem5.1]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid p) amount 6 q) proportion r) percent
water s) amount i) amount 17 t) proportion u) percent
v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 4
Quantity? s
Value? 24

OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid b) amount 3 q) proportion r) percent
water e) amount 7 f) proportion g) percent
i) amount 3 j) proportion 0.15 k) percent

acid h) amount-15%-solution 20.0
water l) amount 17 m) proportion n) percent

acid p) amount 6 q) proportion r) percent
water s) amount i) amount 17 t) proportion u) percent
v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 4
Quantity? s
Value? 24

OPERATION VERIFIED TO BE CORRECT
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid  b) amount 3  c) proportion 0.3  d) percent
      water e) amount 7  f) proportion  g) percent

acid  i) amount 3  j) proportion 0.15  k) percent
      water l) amount 17  m) proportion  n) percent

o) amount-mixture
acid  p) amount 6  q) proportion  r) percent
      water s) amount 24  t) proportion  u) percent
      v) full proportion constant 1.0  w) full percent constant 100.0  y) other value

That value doesn't belong with any quantity in this problem.
You used a simplified version of the rule:
THE PROPORTION OF A SUBSTANCE IN A SOLUTION IS THE AMOUNT OF THE SUBSTANCE
IN THE SOLUTION
DIVIDED BY THE AMOUNT OF THE SOLUTION
You must refine your selection of quantities to amount-mixture instead of amount-water-mixture.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?
That value doesn't belong with any quantity in this problem.
You used a simplified version of the rule:
THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE
PROPORTION OF THE
SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION
You must refine your selection of quantities to amount-66%-alloy instead of amount-
second-alloy
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
copper alloy. What is the percentage of copper in the second alloy
if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
copper alloy. What is the percentage of copper in the second alloy
if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
copper alloy. What is the percentage of copper in the second alloy
if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
copper alloy. What is the percentage of copper in the second alloy
if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
copper alloy. What is the percentage of copper in the second alloy
if the mixture contains 66% copper?
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
a) & \quad \text{amount-45\%-alloy} & 80.0 \\
\text{pure-copper} & \quad \text{b) amount} & 36 \\
\text{other-metal} & \quad \text{c) proportion} & 0.45 \\
\text{d) percent} & \quad \text{e) amount} \\
\text{f) proportion} & \quad \text{g) percent} \\
h) & \quad \text{amount-second-alloy} \\
\text{pure-copper} & \quad \text{i) amount} \\
\text{j) proportion} & \quad \text{k) percent} \\
\text{l) amount} & \quad \text{m) proportion} \\
\text{n) percent} & \quad \text{o) amount-66\%-alloy} \\
\text{pure-copper} & \quad \text{p) amount} & 99 \\
\text{q) proportion} & \quad \text{r) percent} \\
\text{other-metal} & \quad \text{s) amount} \\
\text{t) proportion} & \quad \text{u) percent} \\
v) & \quad \text{full proportion constant} & 1.0 \\
w) & \quad \text{full percent constant} & 100.0 \\
y) & \quad \text{other value}
\end{align*}
\]

1) note goal 
2) note constant 
3) specify a variable 
4) give calculated result 
5) give expression 
6) solve 
7) help-hints 
8) help-menu-options 
9) help-algebra 
10) exit-problem

Option? 4 
Quantity? \(i\) 
Value? .63

OPERATION VERIFIED TO BE CORRECT

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
a) & \quad \text{amount-45\%-alloy} & 80.0 \\
\text{pure-copper} & \quad \text{b) amount} & 36 \\
\text{other-metal} & \quad \text{c) proportion} & 0.45 \\
\text{d) percent} & \quad \text{e) amount} \\
\text{f) proportion} & \quad \text{g) percent} \\
h) & \quad \text{amount-second-alloy} \\
\text{pure-copper} & \quad \text{i) amount} \\
\text{j) proportion} & \quad \text{k) percent} \\
\text{l) amount} & \quad \text{m) proportion} \\
\text{n) percent} & \quad \text{o) amount-66\%-alloy} \\
\text{pure-copper} & \quad \text{p) amount} & 99 \\
\text{q) proportion} & \quad \text{r) percent} \\
\text{other-metal} & \quad \text{s) amount} \\
\text{t) proportion} & \quad \text{u) percent} \\
v) & \quad \text{full proportion constant} & 1.0 \\
w) & \quad \text{full percent constant} & 100.0 \\
y) & \quad \text{other value}
\end{align*}
\]

1) note goal 
2) note constant 
3) specify a variable 
4) give calculated result 
5) give expression 
6) solve 
7) help-hints 
8) help-menu-options 
9) help-algebra 
10) exit-problem

Option? 4 
Quantity? \(j\) 
Value? .9

OPERATION VERIFIED TO BE CORRECT

Installing problem constant 70.0 in amount-second-alloy as you made use of it in computing your result.
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

| a) amount | 45%-alloy | 80.0 |
| pure-copper | b) amount | 36.0 |
| other-metal | c) proportion | 0.45 |
| d) percent | h) amount | second-alloy | 70.0 |
| pure-copper | i) amount | 63.0 |
| other-metal | j) proportion | 0.9 |
| k) percent | l) amount | m) proportion |
| n) percent | o) amount | 66%-alloy |
| pure-copper | p) amount | 99.0 |
| other-metal | q) proportion |
| r) percent | s) amount |
| t) proportion |
| u) percent | v) full proportion constant | 1.0 |
| w) full percent constant | 100.0 |
| y) other value |

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

| a) amount | 25%-solution |
| medicine | b) amount |
| water | c) proportion |
| d) percent |
| h) amount | 5%-solution |
| medicine | i) amount |
| water | j) proportion |
| k) percent |
| o) amount | 12%-solution |
| medicine | p) amount |
| water | q) proportion |
| r) percent |
| v) full proportion constant | 1.0 |
| w) full percent constant | 100.0 |
| y) other value |

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

B.3 Student CP
Menu option q is as yet unspecified. Please enter a value or variable for it: .12
Menu option o is as yet unspecified. Please enter a value or variable for it: .5

EXPRESSION VERIFIED TO BE CORRECT
OPERATION VERIFIED TO BE CORRECT

\((0.25 \times x1) + (0.05 \times x2) = (0.12 \times 5)\)

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{medicine} & \quad \text{b) amount} & \quad \text{c) proportion 0.25} & \quad \text{d) percent} \\
\text{water} & \quad \text{e) amount} & \quad \text{f) proportion} & \quad \text{g) percent} \\
\text{medicine} & \quad \text{i) amount} & \quad \text{j) proportion 0.05} & \quad \text{k) percent} \\
\text{water} & \quad \text{l) amount} & \quad \text{m) proportion} & \quad \text{n) percent} \\
\text{medicine} & \quad \text{p) amount} & \quad \text{q) proportion 0.12} & \quad \text{r) percent} \\
\text{water} & \quad \text{s) amount} & \quad \text{t) proportion} & \quad \text{u) percent} \\
\end{align*}
\]

\(v)\) full proportion constant 1.0 \(w)\) full percent constant 100.0 \(y)\) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 5
Expression? a+h=0
Hit return to continue.

EXPRESSION VERIFIED TO BE CORRECT

\((0.25 \times x1) + (0.05 \times x2) = (0.12 \times 5)\)

\((x1 + x2) = 5\)

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{medicine} & \quad \text{b) amount} & \quad \text{c) proportion 0.25} & \quad \text{d) percent} \\
\text{water} & \quad \text{e) amount} & \quad \text{f) proportion} & \quad \text{g) percent} \\
\text{medicine} & \quad \text{i) amount} & \quad \text{j) proportion 0.05} & \quad \text{k) percent} \\
\text{water} & \quad \text{l) amount} & \quad \text{m) proportion} & \quad \text{n) percent} \\
\text{medicine} & \quad \text{p) amount} & \quad \text{q) proportion 0.12} & \quad \text{r) percent} \\
\text{water} & \quad \text{s) amount} & \quad \text{t) proportion} & \quad \text{u) percent} \\
\end{align*}
\]

\(v)\) full proportion constant 1.0 \(w)\) full percent constant 100.0 \(y)\) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 6

1. \(((0.25 \times x1) + (0.05 \times x2)) = (0.12 \times 5)\)
2. \((x_1 + x_2) = 5\)
Enter a list of numbers in increasing order designating the expressions you want to solve - (1 2)
Would you like to 1) compute the two unknowns 2) remove a nongoal unknown ? 1

Will now solve the equations
\[((0.25 \times x_1) + (0.05 \times x_2)) = (0.12 \times 5)\]
\((x_1 + x_2) = 5\)

Variable \(x_1\) replaced with 1.75
Variable \(x_2\) replaced with 3.25

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?
Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.
Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

- a) amount-20%-solution 10.0
- alcohol b) amount 2 c) proportion 0.2 d) percent
- water e) amount f) proportion g) percent
- v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

Option? 10
Do you want to exit because 1) the problem is solved 2) some other reason? 1
Would you like to 1) stop now 2) load a new problem? 2

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

- a) amount-30%-solution
- acid b) amount c) proportion d) percent
- water e) amount f) proportion g) percent
- h) amount-15%-solution
- acid i) amount j) proportion k) percent
- water l) amount m) proportion n) percent
- o) amount-mixture
- acid p) amount q) proportion r) percent
- water s) amount t) proportion u) percent
- v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

Option? 4
Quantity? p
Value? 6
OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

- a) amount-30%-solution
- acid b) amount c) proportion d) percent
- water e) amount f) proportion g) percent
- h) amount-15%-solution
- acid i) amount j) proportion k) percent
- water l) amount m) proportion n) percent
- o) amount-mixture
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

<table>
<thead>
<tr>
<th>Alcohol</th>
<th>b) Amount</th>
<th>C) Proportion</th>
<th>d) Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Water</td>
<td>e) Amount</td>
<td>f) Proportion</td>
<td>g) Percent</td>
</tr>
<tr>
<td>Alcohol</td>
<td>i) Amount</td>
<td>j) Proportion</td>
<td>k) Percent</td>
</tr>
<tr>
<td>Water</td>
<td>l) Amount</td>
<td>m) Proportion</td>
<td>n) Percent</td>
</tr>
<tr>
<td>Alcohol</td>
<td>p) Amount</td>
<td>q) Proportion</td>
<td>r) Percent</td>
</tr>
<tr>
<td>Water</td>
<td>s) Amount</td>
<td>t) Proportion</td>
<td>u) Percent</td>
</tr>
<tr>
<td>v) Full</td>
<td>Full proportion constant 1.0</td>
<td>w) Full percent constant 100.0</td>
<td>y) Other value</td>
</tr>
</tbody>
</table>

Menu option c is as yet unspecified. Please enter a value or variable for it: x2
Menu option j is as yet unspecified. Please enter a value or variable for it: x1
Menu option q is as yet unspecified. Please enter a value or variable for it: x7

The following equation was not found to be valid:

\[0.2 + x1 = 0.7\]

You used a simplified version of the rule:

THE PROPORTIONS OF ALL THE SUBSTANCES THAT MAKE UP A SOLUTION ADD UP TO THE FULL PROPORTION VALUE 1.0

You must refine your selection of quantities to proportion-water-20%-sol instead of proportion-alcohol-pure-sol.

You used the following ILLEGAL rule:

WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE PROPORTION OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING THE PROPORTIONS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS.

to combine 0.2 of proportion-alcohol-20%-sol with x1 of proportion-alcohol-pure-sol.

This rule is wrong because:

Summing the amounts of these substances is legal (cr8) but it is a mistake to make an analogy
and sum the proportions of these substances. You should monitor your use of computational relationships. When two solutions are mixed, the proportion of a substance in the solution falls between the proportions of the substance in each of the solutions.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

```
a) amount-20%-solution
  alcohol  b) amount  c) proportion 0.2  d) percent
  water    e) amount  f) proportion  g) percent

h) amount-pure-solution
  alcohol  i) amount  j) proportion x1  k) percent
  water    l) amount  m) proportion  n) percent

o) amount-70%-solution
  alcohol  p) amount  q) proportion 0.7  r) percent
  water    s) amount  t) proportion  u) percent
  v) full proportion constant 1.0  w) full percent constant 100.0  y) other value
```

Menu option i is as yet unspecified. Please enter a value or variable for it: x2
Menu option o is as yet unspecified. Please enter a value or variable for it: x3

amount-alcohol-70%-sol does not combine with left operand using equate.
You used a simplified version of the rule:
WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING TOGETHER THE AMOUNTS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS
You must refine your selection of quantities to amount-alcohol-20%-sol instead of proportion-alcohol-20%-sol
The following equation was not found to be valid:
\[(0.2 + x2) = (0.7 * x3)\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

```
a) amount-20%-solution
  alcohol  b) amount  c) proportion 0.2  d) percent
  water    e) amount  f) proportion  g) percent

h) amount-pure-solution
  alcohol  i) amount x2  j) proportion x1  k) percent
  water    l) amount  m) proportion  n) percent

o) amount-70%-solution x3
  alcohol  p) amount  q) proportion 0.7  r) percent
  water    s) amount  t) proportion  u) percent
  v) full proportion constant 1.0  w) full percent constant 100.0  y) other value
```

Option? 5
Expression? c+i=q*o
Hit return to continue.
Expression? \((b+i)/o=q\)
Hit return to continue.

Menu option b is as yet unspecified. Please enter a value or variable for it: 2
Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.
EXPRESSION VERIFIED TO BE CORRECT
\[\left(\frac{2 + x^2}{x^3}\right) = 0.7\]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

- a) amount-20%-solution 10.0
- b) amount 2
- c) proportion 0.2
- d) percent
- e) amount
- f) proportion
- g) percent
- h) amount-pure-solution
- i) amount \(x_2\)
- j) proportion \(x_1\)
- k) percent
- l) amount
- m) proportion
- n) percent
- o) amount-70%-solution \(x_3\)
- p) amount
- q) proportion 0.7
- r) percent
- s) amount
- t) proportion
- u) percent
- v) full proportion constant 1.0
- w) full percent constant 100.0
- y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 10
Do you want to exit because 1) the problem is solved 2) some other reason? 2
Would you like to 1) stop now 2) load a new problem? 1

B.5 Student DH

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

- a) amount-20%-solution
- b) amount
- c) proportion
- d) percent
- e) amount
- f) proportion
- g) percent
- v) full proportion constant 1.0
- w) full percent constant 100.0
- y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 8
which option would you like to have explained? 1, 2, 3, 4, 5, 6, 7, 8, 9, or 10? 1
At various stages in solving a problem there are quantities you want to calculate or solve for.
This option should be used to note to the system which quantities you are trying to obtain. You will be prompted for Quantity? to which you should answer with the letter from the menu of quantities.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol: } 10 \times 0.2 = 2 \text{ gallons}
\]

\[
\text{water: } 10 - 2 = 8 \text{ gallons}
\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{amount-alcohol-20\%-sol} = \text{proportion-alcohol-20\%-sol} \times \text{amount-20\%-solution} \]

Would you like help on:
1) strategies you have not used that you have not used
3) reviewing strategies you have used this domain that you have used
5) return to the main menu

Would you like to see:
1) Specific usages of the rule for this problem (if any)
3) A more specific statement of the rule (if any)
5) More rules that satisfy your last request (if any)
6) a return to the previous menu

THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE PROPORTION OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0

THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE PROPORTION OF THE SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol} = 10 \times 0.2 = 2 \quad \text{gallons of alcohol}
\]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\text{alcohol} = 10 \times 0.7 = 7 \quad \text{gallons of pure alcohol needed}
\]

B.6 Student JH

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{alcohol} = 10 \times 0.2 \]

\[ \text{alcohol} = 2 \text{ gallons} \]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

To solve, we can use the following formula:

\[
\text{Amount of Alcohol} = \text{Volume} \times \text{Proportion}
\]

Given:
- Volume = 10 gallons
- Proportion = 0.2 (20%)

\[
\text{Amount of Alcohol} = 10 \times 0.2 = 2 \text{ gallons}
\]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

To solve, we can use the following formula:

\[
\text{Percentage of Acid} = \frac{\text{Amount of Acid}}{\text{Total Volume}} \times 100
\]

Given:
- Volume of first solution = 10 gallons
- Proportion of acid in first solution = 0.3 (30%)
- Volume of second solution = 20 gallons
- Proportion of acid in second solution = 0.15 (15%)

\[
\text{Amount of Acid} = 10 \times 0.3 + 20 \times 0.15 = 3 + 3 = 6
\]

\[
\text{Percentage of Acid} = \frac{6}{10 + 20} \times 100 = 20\%
\]

Belong with any quantity in this problem. You used the following ILLEGAL rule:

WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE PROPORTION OF A SUBSTANCE IN THE COMPOSITE SOLUTION IS FOUND BY ADDING THE PROPORTIONS OF THE SAME SUBSTANCE IN THE CONSTITUENT SOLUTIONS.

This rule is wrong because:
Summing the amounts of these substances is legal (cr8) but it is a mistake to make an analogy.
and sum the proportions of these substances. You should monitor your use of computational relationships. When two solutions are mixed, the proportion of a substance in the solution falls between the proportions of the substance in each of the solutions.

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
3) reviewing strategies you have used in this domain that you have used
5) return to the main menu

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE PROPORTION OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0

is a rule that has as an input the known entity amount-mixture.

Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2
THE PROPORTION OF A SUBSTANCE IN A SOLUTION IS THE AMOUNT OF THE SUBSTANCE IN THE SOLUTION
DIVIDED BY THE AMOUNT OF THE SOLUTION
is a rule that has as an input the known entity amount-mixture
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2

WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF
ONE OF THE CONSTITUENT SOLUTIONS IS FOUND BY SUBTRACTING THE AMOUNT OF THE OTHER CONSTITUENT
SOLUTION FROM THE AMOUNT OF THE COMPOSITE SOLUTION
is a rule that has as an input the known entity amount-mixture
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2

WHEN TWO CONSTITUENT SOLUTIONS ARE MIXED TO FORM A COMPOSITE SOLUTION, THE AMOUNT OF
ONE OF THE CONSTITUENT SOLUTIONS IS FOUND BY SUBTRACTING THE AMOUNT OF THE OTHER CONSTITUENT
SOLUTION FROM THE AMOUNT OF THE COMPOSITE SOLUTION
is a rule that has as an input the known entity amount-mixture
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2

WHEN TWO SUBSTANCES ARE MIXED TO FORM A SOLUTION, THE AMOUNT OF ONE
SUBSTANCE FROM THE AMOUNT OF THE SOLUTION
is a rule that has as an input the known entity amount-15%-solution
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 1

amount-water-15%-sol( ) = amount-15%-solution( ) - amount-alcohol-15%-sol( )
Would you like to see values with that equation (y or n)? y
amount-water-15%-sol(Unknown) = amount-15%-solution(20.0) - amount-alcohol-15%-sol(Unknown)
Would you like to 1) return to the previous menu, 2) see another example of the strategy? 1

Would you like help on :
1) strategies you have not used 2) computable relationships for this domain
3) reviewing strategies you have used 4) reviewing computable relationships for this domain that you have used
5) return to the main menu 5

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

| acid | b) amount 10.0 |
| water | e) amount |
| c) proportion |
| d) percent |
| f) proportion |
| g) percent |
| h) amount-15%-solution |
| i) amount |
| j) proportion |
| k) percent |
| l) amount |
| m) proportion |
| n) percent |
| o) amount-mixture |
| p) amount |
| q) proportion |
| r) percent |
| s) amount |
| t) proportion |
| u) percent |
| v) full proportion |
| w) full percent |
| x) constant 1.0 |
| y) other value |
| z) other value |

1) note goal result
2) note constant
3) specify a variable
4) give calculated
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem
Option? 4
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{acid} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{acid} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{acid} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{acid} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{acid} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{v) full proportion constant} \; 1.0 \\
\text{w) full percent constant} \; 100.0 \\
\text{y) other value} \\
\text{1) note goal} \\
\text{2) note constant} \\
\text{3) specify a variable} \\
\text{4) give calculated result} \\
\text{5) give expression} \\
\text{6) solve} \\
\text{7) help-hints} \\
\text{8) help-menu-options} \\
\text{Option? 10} \\
\text{Do you want to exit because} \; 1) \; \text{the problem is solved} \; 2) \; \text{some other reason} \; ? \; 2 \\
\text{Would you like to} \; 1) \; \text{stop now} \; 2) \; \text{load a new problem} \; ? \; 1
\]

**B.7 Student JS**

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol} \quad \text{amount} \\
\text{water} \quad \text{amount} \\
\text{v) full proportion constant} \; 1.0 \\
\text{w) full percent constant} \; 100.0 \\
\text{y) other value} \\
\text{1) note goal} \\
\text{2) note constant} \\
\text{3) specify a variable} \\
\text{4) give calculated result} \\
\text{5) give expression} \\
\text{6) solve} \\
\text{7) help-hints} \\
\text{8) help-menu-options} \\
\text{Option? 4} \\
\text{Quantity? b} \\
\text{Value? 2} \\
\text{OPERATION VERIFIED TO BE CORRECT} \\
\text{Installing problem constant} \; 0.2 \; \text{in proportion-alcohol-20%-sol as you made use of it in computing your result.} \\
\text{Installing problem constant} \; 10.0 \; \text{in amount-20%-solution as you made use of it in computing your result.} \\
\]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
a) \; \text{amount-20%-solution} \; 10.0
\]
If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

---

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

---

If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

```
acid b) amount 3
water e) amount

a) amount-30%-solution 10.0
c) proportion 0.3
d) percent
f) proportion
g) percent

h) amount-15%-solution
i) amount

m) proportion

k) percent

l) amount

o) amount-mixture
p) amount

q) proportion

r) percent
t) proportion

u) percent

v) full proportion constant 1.0

w) full percent constant 100.0

y) other value
```

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

```
acid b) amount 3
c) proportion 0.3
d) percent
f) proportion
g) percent

h) amount-15%-solution
i) amount

m) proportion

k) percent

l) amount

o) amount-mixture
p) amount

q) proportion

r) percent
t) proportion

u) percent

v) full proportion constant 1.0

w) full percent constant 100.0

y) other value
```

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{acid amount} \quad \text{water amount} \\
\text{acid proportion} \quad \text{water proportion} \\
\text{full proportion} \quad \text{full percent} \\
\text{note goal} \quad \text{note constant} \\
\text{give expression} \quad \text{solve} \\
\text{help-algebra} \quad \text{help-hints} \\
\text{Option? 4} \quad \text{Value? 30} \\
\text{OPERATION VERIFIED TO BE CORRECT}
\]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{acid amount} \quad \text{water amount} \\
\text{acid proportion} \quad \text{water proportion} \\
\text{full proportion} \quad \text{full percent} \\
\text{note goal} \quad \text{note constant} \\
\text{give expression} \quad \text{solve} \\
\text{help-algebra} \quad \text{help-hints} \\
\text{Option? 4} \quad \text{Value? 6} \\
\text{OPERATION VERIFIED TO BE CORRECT}
\]
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\begin{align*}
\text{acid} & \quad b) \quad \text{amount} & \quad 3 & \quad c) \quad \text{proportion} & \quad 0.3 & \quad d) \quad \text{percent} \\
\text{water} & \quad e) \quad \text{amount} & \quad f) \quad \text{proportion} & \quad g) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{acid} & \quad i) \quad \text{amount} & \quad j) \quad \text{proportion} & \quad 0.15 & \quad k) \quad \text{percent} \\
\text{water} & \quad l) \quad \text{amount} & \quad m) \quad \text{proportion} & \quad n) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{acid} & \quad p) \quad \text{amount} & \quad q) \quad \text{proportion} & \quad 0.2 & \quad r) \quad \text{percent} \\
\text{water} & \quad s) \quad \text{amount} & \quad t) \quad \text{proportion} & \quad u) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{v) \ full \ proportion \ constant} & \quad 1.0 & \quad \text{w) \ full \ percent \ constant} & \quad 100.0 & \quad y) \ \text{other \ value} \\
\end{align*}
\]

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
\text{pure-copper} & \quad b) \quad \text{amount} & \quad c) \quad \text{proportion} & \quad d) \quad \text{percent} \\
\text{other-metal} & \quad e) \quad \text{amount} & \quad f) \quad \text{proportion} & \quad g) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{pure-copper} & \quad i) \quad \text{amount} & \quad j) \quad \text{proportion} & \quad k) \quad \text{percent} \\
\text{other-metal} & \quad l) \quad \text{amount} & \quad m) \quad \text{proportion} & \quad n) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{pure-copper} & \quad p) \quad \text{amount} & \quad q) \quad \text{proportion} & \quad r) \quad \text{percent} \\
\text{other-metal} & \quad s) \quad \text{amount} & \quad t) \quad \text{proportion} & \quad u) \quad \text{percent} \\
\end{align*}
\]

\[
\begin{align*}
\text{v) \ full \ proportion \ constant} & \quad 1.0 & \quad \text{w) \ full \ percent \ constant} & \quad 100.0 & \quad y) \ \text{other \ value} \\
\end{align*}
\]
Menu option c is as yet unspecified. Please enter a value or variable for it: 0.45
Menu option a is as yet unspecified. Please enter a value or variable for it: 80
Menu option j is as yet unspecified. Please enter a value or variable for it: x1
Menu option h is as yet unspecified. Please enter a value or variable for it: 70
Menu option q is as yet unspecified. Please enter a value or variable for it: 0.66
You illegally tried to combine amount-copper-66%-alloy and proportion-copper-66%-alloy with an equate operator.
There are no relationships that allow you to legally combine these quantities with equate.
You illegally tried to equate amount-copper-66%-alloy and proportion-copper-66%-alloy.
For two terms to be equated they must compute the same quantity.
The following equation was not found to be valid:

\[(0.45 \times 80) + (x1 \times 70) = 0.66\]

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

```plaintext
pure-copper  b) amount  e) amount
c) proportion 0.45  f) proportion
d) percent  g) percent

other-metal

h) amount-second-alloy 70
i) amount  j) proportion x1
k) percent  l) amount  m) proportion
n) percent

o) amount-66%-alloy
p) amount  q) proportion 0.66
r) percent  s) amount  t) proportion
u) percent
v) full proportion constant 1.0
w) full percent constant 100.0
y) other value
```

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem
Option? 5
Expression? \((c \times a) + (j \times h) = q \times o\)

Menu option o is as yet unspecified. Please enter a value or variable for it: 150
EXPRESSION VERIFIED TO BE CORRECT

\[((0.45 \times 80) + (x1 \times 70)) = (0.66 \times 150)\]

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

```plaintext
pure-copper  b) amount  e) amount
c) proportion 0.45  f) proportion
d) percent  g) percent

other-metal

h) amount-second-alloy 70
i) amount  j) proportion x1
k) percent  l) amount  m) proportion
n) percent

o) amount-66%-alloy 150
p) amount  q) proportion 0.66  r) percent
```

263
Will now solve the equation

\[((0.45 \times 80) + (x_1 \times 70)) = (0.66 \times 150)\]

Variable \(x_1\) replaced with 0.9

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{alcohol} = 0.2 \times 10 \]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{alcohol} = 0.2 \times 10 \]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{alcohol} = 0.2 \times 10 \]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

Will now execute the computation rule

\[ x_1 = (0.2 \times 10) \]

Variable \( x_1 \) replaced with 2.0

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
\text{pure-copper} & \quad b) \text{ amount} \quad c) \text{ proportion} \quad d) \text{ percent} \\
\text{other-metal} & \quad e) \text{ amount} \quad f) \text{ proportion} \quad g) \text{ percent} \\
\text{amount-45%-alloy} & \quad h) \text{ amount-second-alloy} \\
\text{pure-copper} & \quad i) \text{ amount} \quad j) \text{ proportion} ** \quad k) \text{ percent} \\
\text{other-metal} & \quad l) \text{ amount} \quad m) \text{ proportion} \quad n) \text{ percent} \\
\text{amount-second-alloy} & \quad o) \text{ amount-66%-alloy} \\
\text{pure-copper} & \quad p) \text{ amount} \quad q) \text{ proportion} \quad r) \text{ percent} \\
\text{other-metal} & \quad s) \text{ amount} \quad t) \text{ proportion} \quad u) \text{ percent} \\
\text{amount-66%-alloy} & \quad v) \text{ full proportion constant} 1.0 \\
\text{full percent constant} & \quad w) \text{ full percent constant} 100.0 \\
\text{other value} & \quad y) \text{ other value} \\
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 4
Quantity? o
Value? 150

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
\text{pure-copper} & \quad b) \text{ amount} \quad c) \text{ proportion} \quad d) \text{ percent} \\
\text{other-metal} & \quad e) \text{ amount} \quad f) \text{ proportion} \quad g) \text{ percent} \\
\text{amount-45%-alloy} & \quad h) \text{ amount-second-alloy} \\
\text{pure-copper} & \quad i) \text{ amount} \quad j) \text{ proportion} ** \quad k) \text{ percent} \\
\text{other-metal} & \quad l) \text{ amount} \quad m) \text{ proportion} \quad n) \text{ percent} \\
\text{amount-second-alloy} & \quad o) \text{ amount-66%-alloy} \\
\text{pure-copper} & \quad p) \text{ amount} \quad q) \text{ proportion} \quad r) \text{ percent} \\
\text{other-metal} & \quad s) \text{ amount} \quad t) \text{ proportion} \quad u) \text{ percent} \\
\text{amount-66%-alloy} & \quad v) \text{ full proportion constant} 1.0 \\
\text{full percent constant} & \quad w) \text{ full percent constant} 100.0 \\
\text{other value} & \quad y) \text{ other value} \\
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 4
Quantity? b
Value? 36

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?
Installing problem constant 0.45 in proportion-copper-45%-alloy as you made use of it in computing your result.

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
\text{pure-copper} & \quad \text{b)} \quad \text{amount} \quad 36 \quad \text{c)} \quad \text{proportion} \quad 0.45 \quad \text{d)} \quad \text{percent} \\
\text{other-metal} & \quad \text{e)} \quad \text{amount} \quad \text{f)} \quad \text{proportion} \quad \text{g)} \quad \text{percent} \\
\text{pure-copper} & \quad \text{i)} \quad \text{amount} \quad \text{j)} \quad \text{proportion} \quad \text{**} \quad \text{k)} \quad \text{percent} \\
\text{other-metal} & \quad \text{l)} \quad \text{amount} \quad \text{m)} \quad \text{proportion} \quad \text{n)} \quad \text{percent} \\
\text{pure-copper} & \quad \text{o)} \quad \text{amount} \quad \text{p)} \quad \text{proportion} \quad \text{q)} \quad \text{percent} \\
\text{other-metal} & \quad \text{r)} \quad \text{amount} \quad \text{s)} \quad \text{proportion} \quad \text{t)} \quad \text{percent} \\
& \quad \text{u)} \quad \text{full proportion constant} \quad 1.0 \quad \text{w)} \quad \text{full percent constant} \quad 100.0 \quad \text{y)} \quad \text{other value}
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem


OPERATION VERIFIED TO BE CORRECT
Installing problem constant 0.66 in proportion-copper-66%-alloy as you made use of it in computing your result.

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[
\begin{align*}
\text{pure-copper} & \quad \text{b)} \quad \text{amount} \quad 36 \quad \text{c)} \quad \text{proportion} \quad 0.45 \quad \text{d)} \quad \text{percent} \\
\text{other-metal} & \quad \text{e)} \quad \text{amount} \quad \text{f)} \quad \text{proportion} \quad \text{g)} \quad \text{percent} \\
\text{pure-copper} & \quad \text{i)} \quad \text{amount} \quad \text{j)} \quad \text{proportion} \quad \text{**} \quad \text{k)} \quad \text{percent} \\
\text{other-metal} & \quad \text{l)} \quad \text{amount} \quad \text{m)} \quad \text{proportion} \quad \text{n)} \quad \text{percent} \\
\text{pure-copper} & \quad \text{o)} \quad \text{amount} \quad \text{p)} \quad \text{proportion} \quad \text{q)} \quad \text{percent} \\
\text{other-metal} & \quad \text{r)} \quad \text{amount} \quad \text{s)} \quad \text{proportion} \quad \text{t)} \quad \text{percent} \\
& \quad \text{u)} \quad \text{full proportion constant} \quad 1.0 \quad \text{w)} \quad \text{full percent constant} \quad 100.0 \quad \text{y)} \quad \text{other value}
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

Option? 4 Quantity? i Value? 63

OPERATION VERIFIED TO BE CORRECT

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

\[ a) \text{ amount-45\%-alloy } 80.0 \]
\[ \text{ pure-copper } \quad b) \text{ amount } 36 \quad c) \text{ proportion } 0.45 \quad d) \text{ percent } \]
\[ \text{ other-metal } \quad e) \text{ amount } \quad f) \text{ proportion } \quad g) \text{ percent } \]
\[ h) \text{ amount-second-alloy } 70.0 \]
\[ \text{ pure-copper } \quad i) \text{ amount } 63 \quad j) \text{ proportion } **0.9 \quad k) \text{ percent } \]
\[ \text{ other-metal } \quad l) \text{ amount } \quad m) \text{ proportion } \quad n) \text{ percent } \]
\[ o) \text{ amount-66\%-alloy } 150 \]
\[ \text{ pure-copper } \quad p) \text{ amount } 99 \quad q) \text{ proportion } 0.66 \quad r) \text{ percent } \]
\[ \text{ other-metal } \quad s) \text{ amount } \quad t) \text{ proportion } \quad u) \text{ percent } \]
\[ v) \text{ full proportion constant } 1.0 \quad w) \text{ full percent constant } 100.0 \quad y) \text{ other value } \]

Option? 4

Quantity? j
Value? .9

OPERATION VERIFIED TO BE CORRECT

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[ a) \text{ amount-25\%-solution} \]
\[ \text{ medicine } \quad b) \text{ amount } \quad c) \text{ proportion } \quad d) \text{ percent } \]
\[ \text{ water } \quad e) \text{ amount } \quad f) \text{ proportion } \quad g) \text{ percent } \]
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
((0.25 \times x_1) + (0.05 \times x_2)) = (0.12 \times 5)
\]

EXPRESSION VERIFIED TO BE CORRECT

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?
B.9 Student KL

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

- alcohol
- water
- amount-20%-solution
- amount
- proportion
- percent
- a) amount-20%-solution
- b) amount
- c) proportion
- d) percent
- v) full proportion constant 1.0
- w) full percent constant 100.0
- y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 4
Quantity? b
Value? 2

OPERATION VERIFIED TO BE CORRECT
Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.
Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

- alcohol
- water
- amount-20%-solution
- amount
- proportion
- percent
- a) amount-20%-solution
- b) amount
- c) proportion
- d) percent
- v) full proportion constant 1.0
- w) full percent constant 100.0
- y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 10

Do you want to exit because 1) the problem is solved 2) some other reason? 1
Would you like to 1) stop now 2) load a new problem? 2
[load /usr1/bayley/tutor/solver/problems/mix-problem2.1]

If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

- alcohol
- water
- amount-in-solution
- amount
- proportion
- percent
- a) amount-in-solution
- b) amount
- c) proportion
- d) percent
- v) full proportion constant 1.0
- w) full percent constant 100.0
- y) other value

1) note goal
2) note constant
3) specify a variable
4) give calculated result
5) give expression
6) solve
7) help-hints
8) help-menu-options
9) help-algebra
10) exit-problem

Option? 4
Quantity? a
Value? 15

OPERATION VERIFIED TO BE CORRECT
If you mix 10 gallons of water with 5 gallons of alcohol, what will be the amount of the solution?

a) amount-in-solution 15
b) amount 5.0
c) proportion
d) percent
e) amount 10.0
f) proportion
g) percent
v) full proportion constant 1.0
w) full percent constant 100.0
y) other value

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

a) amount-45%-alloy
b) amount
c) proportion
d) percent
e) amount
f) proportion
g) percent

h) amount-second-alloy
i) amount
j) proportion
k) percent
l) amount
m) proportion
n) percent

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

a) amount-45%-alloy
b) amount
c) proportion
d) percent
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?

A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?
A man mixed 80 pounds of a 45% copper alloy with 70 pounds of a second copper alloy. What is the percentage of copper in the second alloy if the mixture contains 66% copper?
B.10 Student LE

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

alcohol b) amount 10 c) proportion d) percent
water e) amount f) proportion g) percent

a) amount-20%-solution 10

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

alcohol b) amount 10 c) proportion d) percent
water e) amount f) proportion g) percent

a) amount-20%-solution 10

Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[ \text{alcohol} \times 0.20 = \text{gallons of alcohol} \]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[ \text{acid} \times 0.30 + \text{acid} \times 0.15 = \text{total acid} \]

\[ \frac{\text{total acid}}{\text{total volume}} \times 100 = \text{percentage of acid} \]
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\begin{align*}
\text{acid} & \quad b) \text{ amount } 3 \\
\text{water} & \quad e) \text{ amount } 7 \\
\text{acid} & \quad i) \text{ amount } 3 \\
\text{water} & \quad l) \text{ amount } 17 \\
\text{acid} & \quad p) \text{ amount } \\
\text{water} & \quad s) \text{ amount } \\
\text{alcohol} & \quad q) \text{ proportion } 0.3 \\
\text{water} & \quad s) \text{ amount } \\
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 4
Quantity? 17
Value? 17

OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\begin{align*}
\text{acid} & \quad b) \text{ amount } 3 \\
\text{water} & \quad e) \text{ amount } 7 \\
\text{acid} & \quad i) \text{ amount } 3 \\
\text{water} & \quad l) \text{ amount } 17 \\
\text{acid} & \quad p) \text{ amount } \\
\text{water} & \quad s) \text{ amount } \\
\text{alcohol} & \quad q) \text{ proportion } 0.3 \\
\text{water} & \quad s) \text{ amount } \\
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 4
Quantity? s
Value? 24

OPERATION VERIFIED TO BE CORRECT
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

- amount-30%-solution: 10.0
- proportion-30%: 0.3
- percent-30%: 30
- amount-15%-solution: 20.0
- proportion-15%: 0.15
- percent-15%: 15
- amount-mixture: 30
- proportion-mixture: 0.2
- percent-mixture: 20

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

- amount-30%-solution: 10.0
- proportion-30%: 0.3
- percent-30%: 30
- amount-15%-solution: 20.0
- proportion-15%: 0.15
- percent-15%: 15
- amount-mixture: 30
- proportion-mixture: 0.2
- percent-mixture: 20
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

A) amount-20%-solution

B) amount

C) proportion

D) percent

E) amount 8

F) proportion

G) percent

H) amount-pure-solution

I) amount

J) proportion

K) percent

L) amount

M) proportion

N) percent

O) amount-70%-solution

P) amount

Q) proportion

R) percent

S) amount

T) proportion

U) percent

V) full proportion constant 1.0

W) full percent constant 100.0

Y) other value

1) note goal

2) note constant

3) specify a variable

4) give calculated result

5) give expression

6) solve

7) help-hints

8) help-menu-options

9) help-algebra

10) exit-problem

Option? 4

Quantity? e

Value? 8

OPERATION VERIFIED TO BE CORRECT

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\begin{align*}
\text{alcohol} & \quad \text{water} \\
a) & \quad \text{b)} & \quad \text{c)} & \quad \text{d)} & \quad \text{e)} & \quad \text{f)} & \quad \text{g)} \\
\text{amount-20%-solution} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} \\
\text{alcohol} & \quad \text{water} & \quad \text{h)} & \quad \text{i)} & \quad \text{j)} & \quad \text{k)} & \quad \text{l)} & \quad \text{m)} & \quad \text{n)} \\
\text{amount-pure-solution} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} \\
\text{alcohol} & \quad \text{water} & \quad \text{o)} & \quad \text{p)} & \quad \text{q)} & \quad \text{r)} & \quad \text{s)} & \quad \text{t)} & \quad \text{u)} \\
\text{amount-70%-solution} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} & \quad \text{amount} & \quad \text{proportion} & \quad \text{percent} \\
\text{alcohol} & \quad \text{water} & \quad \text{v)} & \quad \text{w)} & \quad \text{x)} & \quad \text{y)} & \quad \text{z)} & \quad \text{other value} \\
\text{full proportion constant 1.0} & \quad \text{full percent constant 100.0} & \quad \text{other value} \\
\end{align*}
\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\text{alcohol} \quad b) \text{ amount} \quad c) \text{ proportion} \quad d) \text{ percent} \\
\text{water} \quad e) \text{ amount} \quad f) \text{ proportion} \quad g) \text{ percent}
\]

\[
\text{alcohol} \quad i) \text{ amount} \quad j) \text{ proportion} \quad k) \text{ percent} \\
\text{water} \quad l) \text{ amount} \quad m) \text{ proportion} \quad n) \text{ percent}
\]

\[
\text{alcohol} \quad p) \text{ amount } x_1 \quad q) \text{ proportion } 0.7 \quad r) \text{ percent} \\
\text{water} \quad s) \text{ amount } 8 \quad t) \text{ proportion } 0.3 \quad u) \text{ percent}
\]

v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 4
Quantity? t
Value? .3

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\text{alcohol} \quad b) \text{ amount} \quad c) \text{ proportion} \quad d) \text{ percent} \\
\text{water} \quad e) \text{ amount} \quad f) \text{ proportion} \quad g) \text{ percent}
\]

\[
\text{alcohol} \quad i) \text{ amount} \quad j) \text{ proportion} \quad k) \text{ percent} \\
\text{water} \quad l) \text{ amount} \quad m) \text{ proportion} \quad n) \text{ percent}
\]

\[
\text{alcohol} \quad p) \text{ amount } x_1 \quad q) \text{ proportion } 0.7 \quad r) \text{ percent} \\
\text{water} \quad s) \text{ amount } 8 \quad t) \text{ proportion } 0.3 \quad u) \text{ percent}
\]

v) full proportion constant 1.0 w) full percent constant 100.0 y) other value

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Expression? q/t=p/s
Hit return to continue.

The following equation was not found to be valid:
\[
\frac{0.7}{0.3} = \frac{x_1}{8}
\]
You used a simplified version of the rule:
THE AMOUNT OF A SOLUTION CAN BE FOUND BY DIVIDING THE AMOUNT OF ONE SUBSTANCE IN THE SOLUTION BY THE PROPORTION OF THAT SUBSTANCE IN THE SOLUTION.
You must refine your selection of quantities to proportion-alcohol-70%-sol instead of amount-water-70%-sol. You used a simplified version of the rule:

**THE PROPORTION OF A SUBSTANCE IN A SOLUTION IS THE AMOUNT OF THE SUBSTANCE IN THE SOLUTION DIVIDED BY THE AMOUNT OF THE SOLUTION**

You must refine your selection of quantities to amount-70%-solution instead of amount-water-70%-sol. You used a simplified version of the rule:

**THE AMOUNT OF A SOLUTION CAN BE FOUND BY DIVIDING THE AMOUNT OF ONE SUBSTANCE IN THE SOLUTION BY THE PROPORTION OF THAT SUBSTANCE IN THE SOLUTION**

You must refine your selection of quantities to amount-water-70%-sol instead of proportion-alcohol-70%-sol.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\frac{\text{amount of alcohol}}{\text{amount of solution}} = \frac{\text{amount of pure alcohol}}{\text{amount of solution}}
\]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\frac{\text{x}}{10} = \frac{8}{0.3}
\]

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\frac{\text{xl}}{0.7} = \frac{8}{0.3}
\]

Variable xl replaced with 18.66666666666667

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

Would you like to 1) stop now 2) load a new problem? 2
[load /usr1/bayley/tutor/solver/problems/mix-problem7.1]

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options Option? 2 Quantity? .25 Value? .25
OPERATION VERIFIED TO BE CORRECT

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

Would you like to 1) stop now 2) load a new problem? 2
[load /usr1/bayley/tutor/solver/problems/mix-problem7.1]
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{medicine} & \quad \text{b)} \quad \text{amount} & \quad \text{c)} \quad \text{proportion} \quad 0.25 & \quad \text{d)} \quad \text{percent} \\
\text{water} & \quad \text{e)} \quad \text{amount} & \quad \text{f)} \quad \text{proportion} \quad 0.75 & \quad \text{g)} \quad \text{percent} \\
\text{h)} \quad \text{amount-5%-solution} & \\
\text{medicine} & \quad \text{i)} \quad \text{amount} & \quad \text{j)} \quad \text{proportion} & \quad \text{k)} \quad \text{percent} \\
\text{water} & \quad \text{l)} \quad \text{amount} & \quad \text{m)} \quad \text{proportion} & \quad \text{n)} \quad \text{percent} \\
\text{o)} \quad \text{amount-12%-solution} & \\
\text{medicine} & \quad \text{p)} \quad \text{amount} & \quad \text{q)} \quad \text{proportion} & \quad \text{r)} \quad \text{percent} \\
\text{water} & \quad \text{s)} \quad \text{amount} & \quad \text{t)} \quad \text{proportion} & \quad \text{u)} \quad \text{percent} \\
\text{v)} \quad \text{full proportion constant 1.0} & \quad \text{w)} \quad \text{full percent constant 100.0} & \quad \text{y)} \quad \text{other value}
\end{align*}
\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

If you mix 10 gallons of water with 5 gallons of alcohol, what will
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[ \text{amount-30\%-solution} = 10 \text{ gallons} \times 0.3 = 3 \text{ gallons} \]

\[ \text{amount-15\%-solution} = 20 \text{ gallons} \times 0.15 = 3 \text{ gallons} \]

\[ \text{amount-mixture} = 10 + 20 = 30 \text{ gallons} \]

\[ \text{percent} = \frac{3 + 3}{30} \times 100 = 20\% \]

You selected the give result option for a quantity that was given in the problem. That value belongs to quantity proportion-alcohol-30\%-sol.
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
v) full proportion constant 1.0  w) full percent constant 100.0  y) other value
A rule applicable to your request is: fd3.1
if THERE IS A RULE WITH KNOWN VALUES FOR ALL ITS INPUTS
then MAKE AN EQUATION FROM THE RULE AND THE OTHER PROBLEM VALUES OR
UNKNOWNs
Would you like to see:
1) Specific usages of the rule for this problem (if any)  2) A more general statement
of the rule (if any)
3) A more specific statement of the rule (if any)  4) An analogous rule (if any)
5) More rules that satisfy your last request (if any)  6) a return to the previous
menu
1
THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE
PROPORTION
OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0
is a rule that has as input the known entities 1.0 & proportion-alcohol-15%-sol
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2
THE PROPORTION OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY SUBTRACTING THE
PROPORTION
OF THE OTHER SUBSTANCE IN THE SOLUTION FROM THE FULL PROPORTION VALUE 1.0
is a rule that has as input the known entities 1.0 & proportion-alcohol-30%-sol
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 2
THE AMOUNT OF A SUBSTANCE IN A SOLUTION CAN BE FOUND BY MULTIPLYING THE
SUBSTANCE IN THE SOLUTION TIMES THE AMOUNT OF THE SOLUTION
is a rule that has as input the known entities proportion-alcohol-15%-sol & amount-15%-sol
Would you like to 1) pursue the use of this rule on the problem? 2) continue as before? 1
amount-alcohol-15%-sol( ) = proportion-alcohol-15%-sol( ) * amount-15%-solution( )
Would you like to see values with that equation (y or n)? y
amount-alcohol-15%-sol(Unknown) = proportion-alcohol-15%-sol(0.15) * amount-15%-solution(20)
Would you like to 1) return to the previous menu, 2) see another example of the strategy? 2
There are no specialization relationships between fd3.1 and other rules for viewing.
Would you like to 1) have a chance to follow up on the same rule, 2) go on to a new rule, 3)
return to previous menu? 3
Would you like help on :
1) strategies you have not used  2) computable relationships for this domain
that you have not used  4) reviewing computable relationships for
3) reviewing strategies you have used this domain that you have used
5) return to the main menu  ? 5

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid
solution, what is the percentage of acid in the mixture?

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U |
| a) | amount-30%-solution | c) | proportion 0.3 | d) | percent |
| b) | amount | f) | proportion | g) | percent |
| e) | amount | h) | amount-15%-solution 20 |
| i) | amount | j) | proportion 0.15 | k) | percent |
| l) | amount | m) | proportion | n) | percent |
| o) | amount-mixture |
| p) | amount | q) | proportion | r) | percent |
| s) | amount | t) | proportion | u) | percent |
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol} \quad \text{b) amount} \quad \text{c) proportion} \quad \text{d) percent}\\
\text{water} \quad \text{e) amount} \quad \text{f) proportion} \quad \text{g) percent}\\
\text{v) full proportion constant 1.0} \quad \text{w) full percent constant 100.0} \quad \text{y) other value}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 10
Do you want to exit because 1) the problem is solved 2) some other reason? 2
Would you like to 1) stop now 2) load a new problem? 1

B.12 Student TW

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol} \quad \text{b) amount} \quad \text{c) proportion} \quad \text{d) percent}\\
\text{water} \quad \text{e) amount} \quad \text{f) proportion} \quad \text{g) percent}\\
\text{v) full proportion constant 1.0} \quad \text{w) full percent constant 100.0} \quad \text{y) other value}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 4
Quantity? b
Value? 2
OPERATION VERIFIED TO BE CORRECT
Installing problem constant 0.2 in proportion-alcohol-20%-sol as you made use of it in computing your result.
Installing problem constant 10.0 in amount-20%-solution as you made use of it in computing your result.

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of alcohol is in the vat?

\[
\text{alcohol} \quad \text{b) amount} \quad \text{c) proportion} \quad \text{d) percent}\\
\text{water} \quad \text{e) amount} \quad \text{f) proportion} \quad \text{g) percent}\\
\text{v) full proportion constant 1.0} \quad \text{w) full percent constant 100.0} \quad \text{y) other value}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem

Option? 10
Do you want to exit because 1) the problem is solved 2) some other reason? 1
Would you like to 1) stop now 2) load a new problem? 2
[load /usr1/bayley/tutor/solver/problems/mix-problem5.1]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{acid} \quad \text{b) amount} \quad \text{c) proportion} \quad \text{d) percent}\\
\text{water} \quad \text{e) amount} \quad \text{f) proportion} \quad \text{g) percent}
\]
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

\[
\text{Acid} \quad b) \text{amount 3} \quad c) \text{proportion 0.3} \quad d) \text{percent}
\]
\[
\text{Water} \quad e) \text{amount 7} \quad f) \text{proportion 0.7} \quad g) \text{percent}
\]

\[
\text{Acid} \quad i) \text{amount} \quad j) \text{proportion} \quad k) \text{percent}
\]
\[
\text{Water} \quad l) \text{amount} \quad m) \text{proportion} \quad n) \text{percent}
\]

\[
\text{Acid} \quad p) \text{amount} \quad q) \text{proportion} \quad r) \text{percent}
\]
\[
\text{Water} \quad s) \text{amount} \quad t) \text{proportion} \quad u) \text{percent}
\]

\[
\text{v) full proportion constant 1.0} \quad \text{w) full percent constant 100.0} \quad \text{y) other value}
\]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid  b) amount 3  c) proportion 0.3  d) percent
water e) amount 7  f) proportion 0.7  g) percent

h) amount-15%-solution 20.0

acid  i) amount 3  j) proportion 0.15  k) percent
water l) amount 17  m) proportion 0.85  n) percent

o) amount-mixture 30

acid  p) amount  q) proportion **  r) percent
water s) amount  t) proportion  u) percent

v) full proportion constant 1.0  w) full percent constant 100.0  y) other value

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem

Option? 4
Quantity? 1
Value? 17

OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid  b) amount 3  c) proportion 0.3  d) percent
water e) amount 7  f) proportion 0.7  g) percent

h) amount-15%-solution 20.0

acid  i) amount 3  j) proportion 0.15  k) percent
water l) amount 17  m) proportion 0.85  n) percent

o) amount-mixture 30

acid  p) amount  q) proportion **  r) percent
water s) amount  t) proportion  u) percent

v) full proportion constant 1.0  w) full percent constant 100.0  y) other value

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem

Option? 4
Quantity? p
Value? 6

OPERATION VERIFIED TO BE CORRECT

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

acid  b) amount 3  c) proportion 0.3  d) percent
water e) amount 7  f) proportion 0.7  g) percent
Which quantity is variable \( x_1 \) in 
\[
((0.3 \times 10.0) + (0.15 \times 20.0)) = (x_1 \times 30)
\]

**EXPRESSION VERIFIED TO BE CORRECT**
\[
((0.3 \times 10.0) + (0.15 \times 20.0)) = (x_1 \times 30)
\]

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?
Will now solve the equation

\[ ((0.3 \times 10.0) + (0.15 \times 20.0)) = (x_1 \times 30) \]

Variable \( x_1 \) replaced with 0.2

If you mix 10 gallons of a 30% acid solution with 20 gallons of a 15% acid solution, what is the percentage of acid in the mixture?

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?
of pure alcohol must be added to produce a 70% solution?

\[
\begin{align*}
\text{alcohol} & \quad \text{b) amount} & \quad \text{c) proportion} & \quad \text{d) percent} \\
\text{water} & \quad \text{e) amount} & \quad \text{f) proportion} & \quad \text{g) percent} \\
\text{h) amount-pure-solution} & \\
\text{alcohol} & \quad \text{i) amount} & \quad \text{j) proportion} & \quad \text{k) percent} \\
\text{water} & \quad \text{l) amount} & \quad \text{m) proportion} & \quad \text{n) percent} \\
o) amount-70\%-solution & \\
\text{alcohol} & \quad \text{p) amount} & \quad \text{q) proportion} & \quad \text{r) percent} \\
\text{water} & \quad \text{s) amount} & \quad \text{t) proportion} & \quad \text{u) percent} \\
v) full proportion constant 1.0 & w) full percent constant 100.0 & y) other value
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

Operation verified to be correct

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\begin{align*}
\text{alcohol} & \quad \text{b) amount} & \quad \text{c) proportion} & \quad \text{d) percent} \\
\text{water} & \quad \text{e) amount} & \quad \text{f) proportion} & \quad \text{g) percent} \\
\text{h) amount-pure-solution} & \\
\text{alcohol} & \quad \text{i) amount} & \quad \text{j) proportion} & \quad \text{k) percent} \\
\text{water} & \quad \text{l) amount} & \quad \text{m) proportion} & \quad \text{n) percent} \\
o) amount-70\%-solution & \\
\text{alcohol} & \quad \text{p) amount} & \quad \text{q) proportion} & \quad \text{r) percent} \\
\text{water} & \quad \text{s) amount} & \quad \text{t) proportion} & \quad \text{u) percent} \\
v) full proportion constant 1.0 & w) full percent constant 100.0 & y) other value
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options 9) help-algebra 10) exit-problem

Operation verified to be correct

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\begin{align*}
\text{alcohol} & \quad \text{b) amount} & \quad \text{c) proportion} & \quad \text{d) percent} \\
\text{water} & \quad \text{e) amount} & \quad \text{f) proportion} & \quad \text{g) percent} \\
\text{h) amount-pure-solution} & \\
\text{alcohol} & \quad \text{i) amount} & \quad \text{j) proportion} & \quad \text{k) percent} \\
\text{water} & \quad \text{l) amount} & \quad \text{m) proportion} & \quad \text{n) percent}
\end{align*}
\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

\[
\begin{align*}
alcohol & \quad b) \text{ amount} \\
water & \quad c) \text{ proportion } 0.2 \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad d) \text{ percent} \\
water & \quad e) \text{ amount} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad f) \text{ proportion} \\
water & \quad g) \text{ percent} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad h) \text{ amount-pure-solution } ** \\
water & \quad i) \text{ amount} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad j) \text{ proportion} \\
water & \quad k) \text{ percent} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad l) \text{ amount} \\
water & \quad m) \text{ proportion} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad n) \text{ percent} \\
water & \quad o) \text{ amount-70%-solution} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad p) \text{ amount} \\
water & \quad q) \text{ proportion} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad r) \text{ percent} \\
water & \quad s) \text{ amount} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad t) \text{ proportion} \\
water & \quad u) \text{ percent} \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad v) \text{ full proportion constant } 1.0 \\
water & \quad w) \text{ full percent constant } 100.0 \\
\end{align*}
\]

\[
\begin{align*}
alcohol & \quad x) \text{ other value} \\
\end{align*}
\]
A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

1) note goal 2) note constant 3) specify a variable 4) give calculated result 5) give expression 6) solve 7) help-hints 8) help-menu-options

Option? 2
Quantity? q
Value? .7
OPERATION VERIFIED TO BE CORRECT

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

Option? 4
Quantity? t
Value? .3
OPERATION VERIFIED TO BE CORRECT

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

Expression? c*a+x1=q*(a+x1)
Hit return to continue.
Which quantity is variable $x_1$ in

$((0.2 \times 10) + x_1) = (0.7 \times (10 + x_1))$

EXPRESSION VERIFIED TO BE CORRECT

$((0.2 \times 10) + x_1) = (0.7 \times (10 + x_1))$

A vat contains 10 gallons of a 20% alcohol solution. How many gallons of pure alcohol must be added to produce a 70% solution?

Variable $x_1$ replaced with 16.66666666666667
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{medicine} & \quad \text{water} \\
\text{a)} & \quad \text{amount-25%-solution} ** \\
\text{b)} & \quad \text{amount} \\
\text{c)} & \quad \text{proportion 0.25} \\
\text{d)} & \quad \text{percent} \\
\text{e)} & \quad \text{amount} \\
\text{f)} & \quad \text{proportion} \\
\text{g)} & \quad \text{percent} \\
\text{h)} & \quad \text{amount-5%-solution} \\
\text{i)} & \quad \text{amount} \\
\text{j)} & \quad \text{proportion 0.05} \\
\text{k)} & \quad \text{percent} \\
\text{l)} & \quad \text{amount} \\
\text{m)} & \quad \text{proportion} \\
\text{n)} & \quad \text{percent} \\
\text{o)} & \quad \text{amount-12%-solution} \\
\text{p)} & \quad \text{amount} \\
\text{q)} & \quad \text{proportion} \\
\text{r)} & \quad \text{percent} \\
\text{medicine} & \quad \text{water} \\
\text{s)} & \quad \text{amount} \\
\text{t)} & \quad \text{proportion} \\
\text{u)} & \quad \text{percent} \\
\text{v)} & \quad \text{full proportion constant 1.0} \\
\text{w)} & \quad \text{full percent constant 100.0} \\
\text{x)} & \quad \text{other value}
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem
Option? 1

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{medicine} & \quad \text{water} \\
\text{a)} & \quad \text{amount-25%-solution} ** \\
\text{b)} & \quad \text{amount} \\
\text{c)} & \quad \text{proportion 0.25} \\
\text{d)} & \quad \text{percent} \\
\text{e)} & \quad \text{amount} \\
\text{f)} & \quad \text{proportion} \\
\text{g)} & \quad \text{percent} \\
\text{h)} & \quad \text{amount-5%-solution} \\
\text{i)} & \quad \text{amount} \\
\text{j)} & \quad \text{proportion 0.05} \\
\text{k)} & \quad \text{percent} \\
\text{l)} & \quad \text{amount} \\
\text{m)} & \quad \text{proportion} \\
\text{n)} & \quad \text{percent} \\
\text{o)} & \quad \text{amount-12%-solution} \\
\text{p)} & \quad \text{amount} \\
\text{q)} & \quad \text{proportion} \\
\text{r)} & \quad \text{percent} \\
\text{water} & \quad \text{medicine} \\
\text{s)} & \quad \text{amount} \\
\text{t)} & \quad \text{proportion} \\
\text{u)} & \quad \text{percent} \\
\text{v)} & \quad \text{full proportion constant 1.0} \\
\text{w)} & \quad \text{full percent constant 100.0} \\
\text{x)} & \quad \text{other value}
\end{align*}
\]

1) note goal 2) note constant 3) specify a variable 4) give calculated result
5) give expression 6) solve 7) help-hints 8) help-menu-options
9) help-algebra 10) exit-problem
Option? 2
Quantity? q
Value? .12

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[
\begin{align*}
\text{a)} & \quad \text{amount-25%-solution} **
\end{align*}
\]
A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?
Which quantity is variable $x_1$ in 
\[ ((0.25 \times x_1) + (0.05 \times x_2)) = (0.12 \times 5) \]

Given that $x_2$ is the unknown variable, we can solve for it.

A nurse must administer 5 ounces of a 12% solution of medicine. In stock are a 25% solution and a 5% solution of this medicine. How many ounces of each should she mix to obtain the 5 ounces of a 12% solution?

\[ ((0.25 \times x_1) + (0.05 \times x_2)) = (0.12 \times 5) \]


<table>
<thead>
<tr>
<th></th>
<th>water</th>
<th>e) amount</th>
<th>f) proportion</th>
<th>g) percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>h)</td>
<td>amount-5%-solution **x2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medicine</td>
<td>i) amount</td>
<td>j) proportion 0.05</td>
<td>k) percent</td>
<td></td>
</tr>
<tr>
<td>water</td>
<td>l) amount</td>
<td>m) proportion</td>
<td>n) percent</td>
<td></td>
</tr>
<tr>
<td>o)</td>
<td>amount-12%-solution 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>medicine</td>
<td>p) amount</td>
<td>q) proportion 0.12</td>
<td>r) percent</td>
<td></td>
</tr>
<tr>
<td>water</td>
<td>s) amount</td>
<td>t) proportion</td>
<td>u) percent</td>
<td></td>
</tr>
<tr>
<td>v)</td>
<td>full proportion constant 1.0</td>
<td>w) full percent constant 100.0</td>
<td>y) other value</td>
<td></td>
</tr>
</tbody>
</table>

1) note goal  2) note constant  3) specify a variable  4) give calculated result
5) give expression  6) solve  7) help-hints  8) help-menu-options
9) help-algebra  10) exit-problem

Option? 10
Do you want to exit because 1) the problem is solved  2) some other reason ? 2
Would you like to 1) stop now  2) load a new problem ? 1
Appendix C

Effect of YAPS and its implementation

We chose to use YAPS (Allen, 1983) for the implementation of our system because it offered a combination of rule-based and object-oriented programming. The object-oriented programming component of YAPS allows us to represent our composite data types using Flavors and to have Flavors inherit methods from mixed-in flavors. The representation of our composite data types using Flavors led to the flexibility in applying strategic rules to different domains described above. The rule-based programming component of YAPS allowed us to implement our rule-based model in a straightforward manner.

Unfortunately, YAPS does not satisfactorily support using production rules with our structured data types which are stored as Flavors objects and asserted into the YAPS database. This is because YAPS operates on a database of list structures, thus when flavors objects are asserted into the database their instance variables are not codable as patterns for the discrimination net. Therefore, left-hand side tests (which are interpreted) must be used to determine matches between the instance variables of flavors objects which have been asserted into the database. Because matching does not take place using the discrimination net, but instead is done through interpreted LISP comparisons, the match-resolve-fire cycle of the production system monitor slows down considerably and there is a perceptible wait, for example, when only three rules fire to verify the user correctly entered a problem constant.

Eventually it became obvious that the slowdown due to interpreted tests would have to be remedied. Therefore, we created a mix-in flavor db-utility
(given in Figure C.1) with a method *make-all-facts* that traverses the hierarchically structured flavors objects and creates list-based facts which are asserted into the database and therefore into the discrimination net. Although this technique causes the program to execute faster, it requires that much of the data in the system be stored twice which eventually leads to memory limitation problems. Furthermore, the failure of the rule interpreter to recognize the update of the value of an instance variable, both previously when left-hand side tests were made to compare instance variable values, and currently when list-based facts are asserted into the database thereby allowing the comparisons to be encoded into the discrimination net, creates a problem. The problem is that rules which should fire due to updated information do not fire because the production system monitor does not reconsider a match unless one of the list-based facts is refreshed. To solve this problem we assert a "refresh" fact into the database with a pointer to the flavors object when we update one of its instance variables. Then a single rule can match the fact in the database corresponding to the changed flavors object, delete that fact, and use the method *make-a-fact* given in Figure C.1 to assert a new fact into the database with the updated value.
Figure C.1. Mix-in flavor db-utility and its methods.

The increment operation applied to quantities in the database is a further example of the use of inherited methods in the system. The increment operation is important to our system because we keep a student model.
(discussed in the next section) which maintains a count of the usage of strategic and computation rules. The increment method given in Figure C.1 is invoked by sending the increment message and an access path to a flavors object. The increment message causes the quantity accessed by the access path to be incremented and a fact to be asserted into the database that indicates that the fact containing the flavors object of the quantity incremented is to be refreshed. Therefore, the count of usages of strategic and computation rules can be kept up to date in this single general way.
Appendix D  Extending the interface

In this appendix, we discuss extending the interface to move beyond a model-tracing tutor to an instructional system, and we present further pedagogical principles, which become desirable to incorporate upon extending the interface. The extensions presented here constitute a design for a continuation of the project, but the implementation of this design is beyond the scope of the project. We begin by discussing how Clancey (1979a, 1979b) and Woolf (1984) based the interaction of their tutoring systems on studies of human discourse. We plan to take a slightly different approach by basing the interaction of our system on a structuring of the knowledge that best facilitates its learnability. This structuring of the knowledge will be done by employing felicity conditions, which VanLehn (1987) speculates may be the "optimal information transmission strategy" for the classroom, and by employing the genetic graph, which encodes relationships that represent the genesis of knowledge. Felicity conditions and the genetic graph provide complementary structures that can be exploited to complete an extended interface. The felicity conditions group and order the rules of the problem-solving procedures, whereas the genetic graph individually relates and orders them. The orderings of the two methods combine to give a more complete, if not total, instructional ordering of the rules and procedures. Having developed this basis for the interaction, we present pedagogical principles for instruction and outline the structuring of our knowledge base that employs felicity conditions and the genetic graph.
D.1. Carrying on an interaction

An intelligent computer-assisted-instructional system must interact with students. Unfortunately, because computer processing of speech and writing is limited, this interaction must take a different form from the manner in which a student employs speech and writing to interact with a human tutor. Without verbal input, the computer must depend on typewritten input, which is too time consuming for a student to generate and too difficult to process in general. In the sections below, we discuss efforts by Clancey (1979a, 1979b) and Woolf (1984) to provide a human-like interaction under these restrictive conditions. Then we present a contrasting approach in which the primary challenge in carrying on an interaction with a student is to provide the proper information at the proper time, instead of providing a duplication of the human tutoring capability. Because human communication abilities are unobtainable, we will depend on technology that is obtainable to present the instructional material and will focus our efforts on creating the proper ordering of the material and on curriculum construction.

D.1.1. Creating discourse procedures for interaction

Guidon (Clancey, 1979a, 1979b) was the first system that provided an extensive tutorial dialogue with a student. Clancey (1979c) employed studies of discourse done in artificial intelligence research (Bobrow et al., 1977; Bruce, 1975; Deutsch, 1974; Winograd, 1977) to guide the implementation of GUIDON's dialogue capabilities. Winograd's study indicated that for carrying on a dialogue it is desirable to have the following forms of knowledge:
1) Knowledge of discourse patterns that is employed in generating utterances. GUIDON employs "discourse procedures" to direct and focus the case dialogue.

2) Augmented knowledge of the domain that is exploited to assist in carrying on a dialogue by applying discourse procedures.

3) Knowledge of the communication situation (including the student's intentions and knowledge and the tutor's intentions) that is employed in controlling the use of discourse patterns.

To further demonstrate these discourse patterns, Clancey (1979a) gives a dialogue transition diagram that shows the invocation structure in which discourse procedures are applied. These transitions take place based on the tutor's goals and on tutoring principles, such as economical presentation and accounting for incorrect behavior, which determine whether and how the goal will be carried out.

D.1.2. Formalizing discourse transitions

Woolf (1984) analyzed human dialogues for speech patterns and arrived at a taxonomy based on the speakers' goals. She noted the presence of the same patterns for different speakers and employed these patterns as templates for generating a discourse. In a system called Meno-tutor, she extended the concept of Clancey's (1979a) state transition diagram to incorporate levels of discourse planning based on pedagogical, strategic, and tactical concerns. The structure of these states is like that of an augmented transition network (Woods, 1970). The states are traversed on a predetermined, though not fixed, path. To represent high-level transitions observed in human tutoring, Meno-tutor applied "meta-rules" that move the tutor off the predetermined path and, therefore, deviate from the standard control mechanism of an augmented transition network. The separation of the discourse-related concerns from the knowledge base allowed Meno-tutor to produce tutorial dialogue from two different knowledge bases that represent unrelated topics. Meno-tutor
employed annotations in each of the knowledge bases in the form of *Michelin Ratings* (Rissland, 1977) to define the importance of the topics to be discussed, corrected, or questioned.

D.1.3. *Discussing strategy with the student*

Woolf further indicated that a tutor or system must teach those attributes that "distinguish a student's reasoning from that of the expert." Our discussion of our rule-based problem-solving model in Chapter 3 states that the expert differs from the student in solving algebra word problems by having compiled strategic and domain rules. Similarly, Clancey and Letsinger (1984) state the importance of the strategy employed, how experts have compiled this strategic knowledge, and that the strategic knowledge needs to be represented in a domain independent form. They encoded the strategic knowledge into NEOMYCIN, a revision of the MYCIN system (Shortliffe, 1976). Hasling et al. (1984) took on the goal of having NEOMYCIN explain its problem-solving strategies. Their system generated strategic explanations from a representation of strategy that is explicit and abstract. To do this they employed meta-rules for the explicit representation of strategy, and they kept these meta-rules domain-independent to make them abstract.

D.1.4. *Strategic and domain interaction based on learning relations*

Because we represent both strategic and domain knowledge, our system must be capable of conducting an interaction to discuss both types of knowledge. To do this, we will drive our interaction with *felicity conditions* (VanLehn, 1987) while exploiting the explanation power of the genetic graph. Felicity conditions are conventions that people naturally obey during natural
language discourse. VanLehn makes an analogy between natural language discourse and teacher-learner discourse to justify employing the term felicity conditions for the conventions he calls one-disjunct-per-lesson and show-work. One-disjunct-per-lesson means introducing only one subprocedure (a procedure whose conditions of applicability are different from others) in a lesson. The show-work convention refers to two kinds of lessons: 1) normal lessons in which all work is shown and 2) optimization lessons where more efficient ways to solve a problem are shown. Felicity conditions are consistent with our student interaction goals in the following ways:

- In our genetic graph representation, knowledge is related by relations that designate the genesis of knowledge. The one-disjunct-per-lesson condition restricts the introduction of new material to that which is most learnable.
- Felicity conditions aid the learning of procedural tasks and therefore are appropriate for learning the steps necessary to solve algebra word problems.
- Experienced teachers naturally generate lesson sequences obeying felicity conditions (VanLehn, 1983). Tutorial interaction controlled by felicity conditions should, therefore, have a degree of authenticity.
- Formation of the more powerful rules that are applied by experts on procedural tasks takes place by the processes of composition and combination. The show-work convention, which is employed to teach rules formed by composition and combination, supports the goal of having the student attain expert level ability.

To enable tutors to be constructed quickly, Woolf (1984) adopted the goal of generalizing the interface for intelligent tutors. Because certain concepts may not fit properly into such a generalization, we argue that an equally valid goal is to specialize an interface to a domain so that the presentation of all concepts may be optimized.

D.2. Pedagogical principles

Our plans to employ felicity conditions in building our interface will allow the extended system to:
• lay out an entire, coherent sequence of topics to be covered,
• take the initiative, and
• introduce the student to each topic in turn.

Such a capability broadens our system to one that teaches as opposed to one that only provides tutorial assistance on concepts that students attempt to apply. Thus, the fact that new concepts can be introduced and taught makes the system an intelligent teaching system instead of an intelligent tutoring system. In contrast, al-Khorezmi tutors by employing model tracing to react appropriately to the level of processing being performed by a student. It does make some progress towards being a teaching system by employing the genetic graph to provide some coherency and sequencing to the topics.

Anderson et al. (1984) suggest that providing instruction in the problem-solving context is an important cognitive principle to be exploited in the design of computer tutors. Therefore, a workbench for solving algebra word problems makes a good context for teaching how to solve algebra word problems and makes extending the tutor to teach a reasonable objective. In the following sections, we present the pedagogical principles to be incorporated in the teaching system.

D.2.1. Controlling combinatorial possibilities of induction

Mathematical instruction employs example extensively. Although these examples are given with explanation, much of the learning takes place through induction from examples, because the explanation may not be understood, may be forgotten, or may be ignored. Therefore, we propose that the examples be ordered by applying felicity conditions to significantly aid

---

44 They justify this by citing studies by Tulving (1983) and Tulving and Thomson (1973) which showed memory recall is increased when the context of the recall matches the context of the study.
the learner. Because employment of felicity conditions causes the combinatorial possibilities in induction from examples to be controlled, employing felicity conditions causes the learner to be significantly aided in the learning process. Indeed, the employment of felicity conditions makes the learning process "possible as opposed to impossible" (VanLehn, 1987).

Both felicity conditions apply to the instruction of the strategic and computation rules of our model. In the strategic component of our model, the goal is for the student to progress through the levels of competence to reach the expert level. The one-disjunct-per-lesson convention can be employed to demonstrate new strategic rules at the appropriate time, and the show-work convention can be employed to demonstrate the compiling of two or more rules into the expert rules. In the computation rule component of our model, the one-disjunct-per-lesson convention can be employed to demonstrate new computation rules and to apply known computation rules to new domains. The show-work convention can be employed to demonstrate the combination of two or more rules into the combined rules that are typical of advanced problem solvers.

D.2.2. Employ genetic relationships for examples and explanation

Although we did not state earlier that employing genetic relationships for examples and explanation is one of our pedagogical principles, the emphasis this thesis places on the employment of the genetic graph foreshadows the statement of this pedagogical principle. Because we plan to employ felicity conditions to guide the teaching interaction, the explanations afforded us by the genetic graph must be appropriate to the implementation of
this plan. The genetic graph relationships are appropriate in the following ways:

- The specialization relationship, when applied to a general rule that is applicable to a problem, gives an example of the usage of that general rule. Students learn by induction from these examples, and felicity conditions organize these examples into lessons through application of the one-disjunct-per-lesson convention.

- When students correctly solve problems, they specialize general rules to create formulas analogous to the formulas that the teacher employed in his examples.

- A new subprocedure often employs rules analogous to rules that the student has previously learned. These analogies come from refinement/generalization links to either more general or more refined rules.

- The show-work convention is employed to demonstrate both the composition/component and the combination/component links that we extended the genetic graph to include.

D.2.3. Progress/regress appropriately for the student

Progressing or regressing appropriately for the student is an obvious pedagogical principle, but it is more meaningful when a skill is modeled in a manner that explicates the progression from novice to expert behavior. In a tutor based on a model of expert competence only, this pedagogical principle means repetitious teaching of the expert methodology until the student meets an acceptable level of proficiency before proceeding to the next topic to be tutored. This type of tutor only allows progress to be measured quantitatively—as what percent of expert behavior can be replicated. In contrast, a tutor that exploits a model of the progression from novice to expert behavior enables progress to be measured qualitatively—as progress through the model.

Adjusting the instruction so that it is appropriate for the student being taught means slowing down by giving more examples, providing for more practice, and repeating topics when appropriate. It also means speeding up by
giving fewer examples, providing for less practice, and perhaps skipping topics when appropriate. According to VanLehn:

learning-from-lesson-sequences lies halfway between induction, where the learner does most of the work, and learning-by-being-told, where the teacher does most of the work (1987, p. 7).

Therefore, employing learning-from-lesson-sequences in a teaching system results in instruction that lies halfway between giving lengthy explanation and guiding problem solving, as in learning-by-being-told, and giving examples that combine or ignore felicity conditions, as in learning by induction. This is another dimension on which to adjust the instruction so that it is appropriate for the student being taught. The range of this dimension is similar to the range from either wet sponge or behaviorist learning (learning-by-being-told) to constructivist learning (induction) (Woolf, 1984).

D.2.4. Teach in the context of problem solving

This pedagogical principle will be embedded in the teaching system by default, because the tutoring system, from which it will be derived, already provides an environment in which problem solving is performed. This cognitive principle of Anderson et al. (1984) is in accord with Collins et al. (1987), whose cognitive apprenticeship advocates situating learning in the context of its use to create a "rich web of memorable associations" between concepts and problem-solving contexts.

D.2.5. Emphasize self-monitoring skills

Our current model lacks any representation of self-monitoring skills. Implementation of this pedagogical principle will require that we extend the

\[45\] Chapter 5 discusses evidence that these skills belong in a model that properly represents expert processing and how these skills were left out of our model.
model (and correspondingly the rulebase and database of the program) by adding rules that represent the knowledge necessary to monitor and check the steps that the model proposes as possibilities. Many of these self-monitoring or checking skills are commonsense skills such as checking that the proportion of a substance in a mixture of two solutions has a value between the values of the proportions of the substance in each of the solutions. This type of reasoning has been termed qualitative reasoning (see deKleer and Brown, 1984) although Clancey (1986b) argues that this usage of the term qualitative reasoning is too restricted. Other self-monitoring or checking skills are more formal. For example, to check whether the units of an answer will be correct, one labels each value with its units and then applies cancelling techniques to arrive at the units of the answer.

D.3. Design of a teaching system

Because we want to provide explanation while we present examples, we employ both felicity conditions and the genetic graph to drive our interface. The felicity conditions group rules into subprocedures and specify an order to the teaching of the subprocedures. They do not specify other information that can be exploited for explanation. The genetic graph relates individual rules to provide explanations of examples through its specialization/generalization links and to provide explanations of how new information relates to old information through its other links. It does not group the necessary rules into procedures to solve problems. In this section, we discuss the issues associated with using felicity conditions and the genetic graph to drive the interaction of a teaching system. We present AND-OR graphs showing the structure of the strategic rule set and derive lessons based on the one-disjunct-per-lesson and show-work conventions. Also, we discuss the possibility of dynamically
constructing an AND-OR graph from a production rule set and the implications this would have on the development and testing of instructional systems.

D.3.1. Lessons represented as disjuncts in the AND-OR graph

Under the one-disjunct-per-lesson convention, a lesson introduces one subprocedure. VanLehn defines subprocedures as AND-OR trees structured as follows:

1) A new OR rule that is placed beneath an existing OR goal. The existing OR goal is called the parent.
2) A new AND goal, which is called by the new OR goal.
3) The new AND has one or more rules. Each rule calls a new OR goal that has just one rule. These ORs are merely a convenience. They provide a place for later subprocedures to attach.
4) Each such OR has a single rule that calls some existing AND goal. These existing AND goals are called kids (1987, p. 17).

Whereas VanLehn has the AND-OR graph built by his learning system, our concern is with building a system encoding a model in the AND-OR graph that represents as completely as possible the progression of knowledge from novice to expert. Therefore, our task is not to have the AND-OR graph built as the system learns, but to select subprocedures from the graph for presentation to the student to facilitate his learning.

To demonstrate the selection of a lesson that will follow the one-disjunct-per-lesson convention, we give in Figure D.1 an AND-OR graph encoding a portion of the problem-solving capability of the weak forward-directed strategy.
Nodes with their names enclosed in a box are AND goal nodes and the remaining nodes are OR goal nodes. The links, which are directed downward, below AND goals are for AND rules and the links below OR goals are for OR rules. Write is a primitive operation (and therefore does not need to be learned) that is employed to write down the result of an operation.

Figure D.1. Part of the weak forward-directed strategy in an AND-OR graph.

There are two subprocedures placed beneath the OR goal 1/Solver. Each of the subprocedures is an OR rule that calls on AND goals, which in this case are Solve/Indirect and Solve/Direct. Therefore, the two lessons selected from this graph will teach the use of the weak forward-directed strategy to solve simple problems requiring the use of a single computation rule that either directly computes the goal or has an input that, when solved for, computes the goal. Neither of these possibilities depends on the other, because no nonprimitive goal that appears as an AND goal in one subprocedure is defined in the other subprocedure. This AND-OR graph, derived by employing felicity conditions, does not designate an ordering for the lessons, thus the genetic
graph sophistication links from $FD2.1$ to $FD2.2$ and from $FD3.1$ to $FD3.2$ can be employed to determine which to teach first.

The AND-OR graph given in Figure D.1 represents only a portion of the knowledge in the knowledge base and, therefore, takes the form of a tree. VanLehn (1987) employs cyclic directed graphs to represent repetitive processes. When we extend the graph in Figure D.1 to represent solving problems that require more than one computation rule application (a repetitive process), we get the graph given in Figure D.2. Because $Multi/WFD$ is dependent on $Solver$ in this graph, the felicity conditions require that $Solve/Indirect$ and $Solve/Direct$ be taught before $Multi/WFD$. In contrast, the genetic graph requires, through it sophistication links, that $FD2.1$ be taught before $FD2.2$ and that $FD3.1$ be taught before $FD3.2$, but not whether they should be taught in the context of $Multi/WFD$ versus $Solve/Indirect$ or $Solve/Direct$. Therefore, the combination of the genetic graph with felicity conditions provides a more complete ordering for the instructional material.
Leaf nodes *Solver*, *Solve/Indirect*, and *Solve/Direct* represent cyclic links back to nodes with the same name in the interior of the graph.

Figure D.2. AND-OR graph representing repetitive processing.

The repetitive processing represented by the graph in Figure D.2 can be employed to solve the problem...
A 750 milliliter solution contains 12% alcohol. What is the amount of nonalcohol liquid in this solution?

The solution of this problem would proceed by first selecting the AND goal Multi/WFD which would first call upon Solve/1value and thereafter Solve/Direct to eventually write

amount of alcohol is .12 \times 750,  
.12 \times 750 is 90, and  
amount of alcohol is 90.

AND goal Multi/WFD follows up by calling Solve/Rest which recursively calls on Solver to call 1/Solver followed by Solve/Direct to eventually write

amount of nonalcohol liquid is 750 - 90,  
750 - 90 is 660, and  
amount of nonalcohol liquid is 660.

Figure D.3 gives a completed graph containing all the disjunctions necessary to include all the weak forward-directed and means-ends strategy rules.
Figure D.3. Completed AND-OR graph of strategic rules.
Figure D.3. (continued).
The inclusion of the expert strategy adds no disjuncts to the graph, since the expert strategy rules are compositions of means-ends strategy rules. Because they are compositions of rules already in the graph, the show-work convention governs the use of the expert strategy rules in the lessons. The
show-work convention declares that lessons that show specific steps must precede lessons that hide those steps (also known as optimization lessons). According to VanLehn, this is a naturally occurring manner constraint in mathematical curricula that constrains the amount of search necessary to do induction from examples.

The OR goals are the points in the graph that determine the subprocedures employed in the one-disjunct-per-lesson convention. It is now apparent that the AND goals determine the compositions of rules to be applied in the optimization lessons of the show-work convention. For example, the composition of the rules under either AND goal of the graph in Figure D.1 designates a process that we call spontaneous computation. The subordination of AND goals under other AND goals in the cyclic directed graph determines the ordering of the optimization or hide-work lessons in which subordinate goals must be taught before their superordinate goals are taught. Given the hard-coded AND-OR graph in Figure D.3 for the strategy rules, an instructional designer could lay out a lesson sequence that specifies the order in which the disjuncts are to be taught to implement the one-disjunct-per-lesson convention along with the order in which the conjuncts are to be taught as optimization lessons to implement the show-work convention.

D.3.2. Dynamic generation of the AND-OR graph

Hard-coding the curriculum in an AND-OR graph is not a general solution to the tutoring problem, especially since the current set of strategic rules are undoubtedly not the optimal set. A more general and more "intelligent" solution would apply meta-rules that examine the strategic rule set to determine the placement of the rules in an AND-OR graph. The application of meta-rules to dynamically create the AND-OR graph would allow
experimentation with different sets of strategic rules in an attempt to determine an optimal set of strategic rules. One possible form this experimentation could take would be to test which provides the best teaching tool: adding refinements of current rules to the rule set or removing rules that are refinements of other rules from the rule set to allow their simplifications to work. Another form this experimentation could take would be to test whether the rules need to be divided into their components to give finer detail or whether different or larger compositions of rules are more suitable. These are experiments to determine different facets of knowledge that relate to the proper granularity for the knowledge.

The lesson sequence must be generated dynamically if the AND-OR graph is generated dynamically. Dynamically generating the lesson sequence from the AND-OR graph requires examining the AND-OR graph for dependencies between AND goal nodes and selecting the independent nodes (those which do not call on other nonprimitive AND goals) at the least depth for beginning lessons. For example, the AND goals of Figure D.1 do call upon other nonprimitive AND goals and should be taught first. These independent AND goals should be followed by those AND goals that are at a shallow depth in the breadth-first search of the AND-OR graph that only call upon AND goals that have been taught. For example, the Multi/WFD goal of Figure D.2 should be taught before the other deeper independent AND goals of Figure D.3. Optimization lessons should be determined by the show-work convention and should be inserted in the lesson sequence after their show-work lesson counterparts. Exactly where optimization lessons should be inserted can only be determined after experimentation with the different possibilities and may even change for different learners.
Another possible source of information to be employed in determining the ordering of lessons is the genetic epistemology information that is stored in the genetic graph. Sophistication links could be exploited to help determine the global ordering of strategy lessons. Analogies by specialization could be exploited to determine multiple components of the same lesson. Analogies by refinement could be exploited to determine local groupings of rules.

For meta-rules to dynamically construct the AND-OR graph, the strategic rules must be represented in a form that allows examination of the strategic rules to determine which of them are applied in conjunction and which of them are applied in disjunction. If the action part of one rule will cause the conditional part of another rule to be satisfied, then the rules are to be applied in conjunction. For example strategic rule FD2.1 produces a bound computation rule with an unknown output, strategic rule FD3.1 takes a bound computation rule with an unknown output and computes that output, and strategic rule FD1 takes a computed value and reports the result. Therefore, these rules each cause the condition part of the following rule to be satisfied and are applied in conjunction. If the condition parts of two rules test for the same things but differ on one or more tests, then they are to be applied in disjunction. For example, strategic rules FD2.1, FD2.2, ME2.1, and ME2.2 all look for computation rules to bind, but select the computation rules that they will bind using different conditions. Therefore, these rules are applied in disjunction.

Recently, research on intelligent tutoring system has emphasized constructing authoring systems to aid in the development of intelligent tutors. An implementation of a system that constructs the AND-OR graph from a rule-based model of a domain to develop lesson plans by exploiting VanLehn's
felicity conditions appears to be a profitable approach to constructing authoring systems for intelligent tutors.