INTELLIGENT TUTORING SYSTEMS FOR SKILL ACQUISITION

by

Derek Tannell Green

© Creative Commons 3.0 Attribution-Share Alike License

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF COMPUTER SCIENCE

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2011
As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Derek Tannell Green entitled Intelligent Tutoring Systems for Skill Acquisition and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Date: 5 August 2011

Paul R. Cohen

Date: 5 August 2011

Carole R. Beal

Date: 5 August 2011

Clayton T. Morrison

Date: 5 August 2011

Ian Fasel

Final approval and acceptance of this dissertation is contingent upon the candidate’s submission of the final copies of the dissertation to the Graduate College. I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

Dissertation Director: Paul R. Cohen

Date: 5 August 2011
STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at the University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. This work is licensed under the Creative Commons Attribution-Share Alike 3.0 United States License. To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/3.0/us/ or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

SIGNED: Derek Tannell Green
ACKNOWLEDGEMENTS

There are a number of people I would like to express my gratitude to for their help and support during my time in graduate school.

First I would like to thank my advisor Paul Cohen for bringing me into his exceptional lab and greatly broadening my understanding of the research world. Working in Paul’s lab has given me innumerable learning experiences that could never be had in a less distinguished lab. Thank you.

Thanks also to my committee members, Carole Beal, Clayton Morrison, and Ian Fasel, for their support over the years.

Special thanks to Tom Walsh for coming in like a whirlwind carrying fresh ideas and generally for being a great inspiration.

Thanks to Yu-Han Chang for all of his assistance and his willingness to help from afar.

Many thanks to my office mates and good friends Wesley Kerr and Daniel Hewlett. Wes was always able to immediately extract the core of any idea, predicting failure points and pointing out inessentials, saving me great amounts of time. In addition, I’ll never forget the ping-pong sessions. Thanks to Daniel Hewlett for all of the great conversations during those pre-defense stressful times and for his willingness and ability to give impromptu lectures on various topics. Also, I believe Daniel deserves credit for the invention of dried fruit office bocce.

I would like to thank Bill Spears for all of the great mentoring on work done during my time at the University of Wyoming (including some critical work done at the pool table.)

Thanks to Suranga Hettiarachchi for his inspirational speeches.

I’ll forever be grateful to my brother Gavin Green for his pivotal advice during hard times.

Thanks also to Antons Rebguns, Jeremy Wright, Anh Tran, and to all of those friends I’ve forgotten to mention.

I would also like to thank Cheryl Craddock for helping me keep all of my ducks in a row.

And finally, thanks to Dino Wenino for the Wyoming nature reports, to Chad Lore for not sending that post-card this time, and to John, Mark, & Robbo along with a number of other similar sources of inspiration.
DEDICATION

To Dad.
# TABLE OF CONTENTS

**LIST OF FIGURES** ................................................. 10  
**LIST OF TABLES** ............................................ 12  
**ABSTRACT** .................................................. 13  
**CHAPTER 1 INTRODUCTION** ...................................... 14  
  1.1 The Availability of Education ............................... 14  
  1.2 The Role of a Teacher .................................. 15  
    1.2.1 A Domain Expert ...................................... 15  
    1.2.2 Knowing the Student ................................ 16  
    1.2.3 The Teacher’s Task .................................. 17  
  1.3 Intelligent Tutoring Systems ............................... 18  
    1.3.1 Domain Expertise .................................... 19  
    1.3.2 Modeling the Student ................................ 19  
    1.3.3 The Tutor’s Task .................................... 20  
  1.4 Designing A Skill Teaching Intelligent Tutoring System . . 20  
    1.4.1 Defining Domain Skills And Skill Mastery ............ 21  
    1.4.2 The State Space ...................................... 22  
    1.4.3 The Markov Assumption ............................... 23  
    1.4.4 Dynamic Bayes Net (DBN) ............................ 25  
    1.4.5 Tutoring Domains For Research Purposes ............. 27  
    1.4.6 Training The ITS .................................... 28  
  1.5 Focus Of This Dissertation ................................. 29  
    1.5.1 General Skill Teaching DBN Template For Curriculum Design 29  
    1.5.2 Complete End-to-End System Validation ................ 30  
    1.5.3 Using The ITS As A Testbed For Educational Research .... 31  
    1.5.4 Thesis Statement ................................... 32  
  1.6 Chapter Overview .......................................... 32  
**CHAPTER 2 RELATED WORK** .................................... 34  
  2.1 Human Education .......................................... 34  
  2.2 Intelligent Tutoring Systems .............................. 35  
    2.2.1 Modeling .............................................. 36  
    2.2.2 The Goal Of The ITS ................................ 38  
  2.3 ITS Work Related To Our Own ............................... 38
<table>
<thead>
<tr>
<th>CHAPTER 3</th>
<th>GENERAL DBN TEMPLATE</th>
<th>41</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Domain Requirements</td>
<td>41</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Model Parameters</td>
<td>41</td>
</tr>
<tr>
<td>3.1.2</td>
<td>Training Data</td>
<td>43</td>
</tr>
<tr>
<td>3.2</td>
<td>DBN Structure</td>
<td>44</td>
</tr>
<tr>
<td>3.2.1</td>
<td>On Representing General Skills</td>
<td>45</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Match Nodes</td>
<td>45</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Facts Known Nodes</td>
<td>46</td>
</tr>
<tr>
<td>3.2.4</td>
<td>Aggregate Nodes</td>
<td>47</td>
</tr>
<tr>
<td>3.2.5</td>
<td>Cascade Links</td>
<td>48</td>
</tr>
<tr>
<td>3.2.6</td>
<td>Full DBN Template</td>
<td>50</td>
</tr>
<tr>
<td>3.3</td>
<td>Curriculum Design Planning</td>
<td>50</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Currently</td>
<td>50</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Scaling Up</td>
<td>51</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 4</th>
<th>CASE STUDY: Laff</th>
<th>52</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Domain Description: Laff</td>
<td>52</td>
</tr>
<tr>
<td>4.1.1</td>
<td>Laff: Learn Arithmetic in Finite Fields</td>
<td>52</td>
</tr>
<tr>
<td>4.1.2</td>
<td>Teacher Actions</td>
<td>56</td>
</tr>
<tr>
<td>4.1.3</td>
<td>Evaluation</td>
<td>57</td>
</tr>
<tr>
<td>4.2</td>
<td>Handcrafted Tutors</td>
<td>58</td>
</tr>
<tr>
<td>4.2.1</td>
<td>Random Tutor (RND)</td>
<td>58</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Expert Tutor (EXP)</td>
<td>59</td>
</tr>
<tr>
<td>4.2.3</td>
<td>Derived Tutor (DRV)</td>
<td>60</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Finite State Machine Tutor (FSM)</td>
<td>60</td>
</tr>
<tr>
<td>4.3</td>
<td>Learned Tutor: DBN</td>
<td>60</td>
</tr>
<tr>
<td>4.4</td>
<td>Experiment Details</td>
<td>64</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Protocol</td>
<td>64</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Experiments</td>
<td>65</td>
</tr>
<tr>
<td>4.5</td>
<td>Results/Analysis</td>
<td>68</td>
</tr>
<tr>
<td>4.5.1</td>
<td>Experiment I</td>
<td>68</td>
</tr>
<tr>
<td>4.5.2</td>
<td>Experiment II</td>
<td>70</td>
</tr>
<tr>
<td>4.6</td>
<td>Discussion</td>
<td>73</td>
</tr>
<tr>
<td>4.6.1</td>
<td>General Overview</td>
<td>73</td>
</tr>
<tr>
<td>4.6.2</td>
<td>Lessons Learned</td>
<td>73</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CHAPTER 5</th>
<th>CASE STUDY: Blast</th>
<th>75</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Domain Description: Blast</td>
<td>75</td>
</tr>
<tr>
<td>5.1.1</td>
<td>An Artificial Language</td>
<td>75</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
<td>Page</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
<td>------</td>
</tr>
<tr>
<td>5.1.2</td>
<td>Muq-Duq: The Language Defined</td>
<td>77</td>
</tr>
<tr>
<td>5.1.3</td>
<td>Acquiring Proficiency in Muq-Duq</td>
<td>80</td>
</tr>
<tr>
<td>5.1.4</td>
<td>Teaching Actions</td>
<td>81</td>
</tr>
<tr>
<td>5.1.5</td>
<td>The Tutoring System</td>
<td>83</td>
</tr>
<tr>
<td>5.1.6</td>
<td>Evaluation</td>
<td>84</td>
</tr>
<tr>
<td>5.2</td>
<td>Handcrafted Tutors</td>
<td>85</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Random Tutor (RND)</td>
<td>85</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Expert Tutor (EXP)</td>
<td>87</td>
</tr>
<tr>
<td>5.3</td>
<td>Learned Tutor: DBN</td>
<td>90</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Network Structure</td>
<td>90</td>
</tr>
<tr>
<td>5.3.2</td>
<td>State Space</td>
<td>98</td>
</tr>
<tr>
<td>5.4</td>
<td>Experiment Details</td>
<td>99</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Protocol</td>
<td>99</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Experimental Conditions</td>
<td>101</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Training/Learning the DBN Model</td>
<td>104</td>
</tr>
<tr>
<td>5.5</td>
<td>Results/Analyses</td>
<td>105</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Evaluating the Domain: Can a Tutor Help You Learn Muq-Duq?</td>
<td>105</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Evaluating the Learned Policy: Does the DBN Succeed?</td>
<td>107</td>
</tr>
<tr>
<td>5.5.3</td>
<td>The Effect of Choice on Learning</td>
<td>111</td>
</tr>
<tr>
<td>5.5.4</td>
<td>Gender Differences in BLAST</td>
<td>114</td>
</tr>
<tr>
<td>5.6</td>
<td>Discussion</td>
<td>119</td>
</tr>
<tr>
<td>5.6.1</td>
<td>General Overview</td>
<td>119</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Improving Test Scores</td>
<td>120</td>
</tr>
<tr>
<td>5.6.3</td>
<td>Understand The Gender Difference</td>
<td>121</td>
</tr>
<tr>
<td>5.6.4</td>
<td>Introducing A Human Tutor</td>
<td>123</td>
</tr>
</tbody>
</table>

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

6.1 Future Work

6.1.1 Bringing Mastery into Alignment with Test Scores | 126

6.1.2 Using Virtual Students To Augment Collected Data | 127

6.1.3 Resolving the Gender Mystery | 127

6.1.4 Deployment To A Large Scale ITS | 128

6.2 Parting Comments | 128

APPENDIX A LAFF: FULL EXPERIMENT PROTOCOL

A.1 Introduction | 130

A.2 Method | 131

A.3 Description of the Problems | 133

A.3.1 Ordering | 135
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A.4 Pre-test/Post-test</td>
<td>137</td>
</tr>
<tr>
<td>APPENDIX B BLAST: ADDITIONAL EXPERIMENT PROTOCOL DETAILS</td>
<td>139</td>
</tr>
<tr>
<td>B.1 Verbal Instructions</td>
<td>139</td>
</tr>
<tr>
<td>B.2 A Visual Guide</td>
<td>139</td>
</tr>
<tr>
<td>APPENDIX C IRB DOCUMENTS</td>
<td>152</td>
</tr>
<tr>
<td>C.1 CITI Completion Report</td>
<td>152</td>
</tr>
<tr>
<td>C.2 IRB 2009</td>
<td>152</td>
</tr>
<tr>
<td>C.3 IRB 2010</td>
<td>152</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>165</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

1.1 Intuitive Comparison of MDP vs DBN ................................. 26
3.1 Match Nodes ........................................................................ 46
3.2 Facts Nodes ......................................................................... 47
3.3 Estimating Facts Known ......................................................... 48
3.4 DBN Aggregate Nodes ........................................................... 49
3.5 Cascade Links ....................................................................... 49
3.6 Multi-layer DBN Template ....................................................... 50
4.1 Handcrafted FSM Policy ......................................................... 61
4.2 Laff DBN, truncated .............................................................. 62
4.3 Finite Field Arithmetic “cheat sheet” ..................................... 65
5.1 Skill Dependencies ............................................................... 82
5.2 DBN Basic Structure ............................................................. 92
5.3 DBN Action Matches ............................................................. 94
5.4 Example Excerpt of CPT for $Q\cdot N\cdot Q$ ............................... 95
5.5 Aggregate Nodes ................................................................. 96
5.6 Fact Nodes ........................................................................... 97
5.7 Cascade Links ...................................................................... 97
5.8 Full DBN ............................................................................. 98
5.9 Mean Test Scores ................................................................. 106
5.10 Mean Test Scores (with 95% Confidence Interval Bars) ........ 107
5.11 Reward Value of State Immediately Preceding Each Test (with 95% Confidence Intervals) ............................................. 110
5.12 Reward Value of State Immediately Preceding Each Test ........ 111
A.1 Finite Field Arithmetic “cheat sheet” .................................. 133
B.1 Log In .............................................................................. 140
B.2 Enter Password ................................................................. 140
B.3 Instructions I ...................................................................... 141
B.4 Instructions II ..................................................................... 142
B.5 Alertness Report ............................................................... 142
B.6 First Multiple Choice Problem .......................................... 143
B.7 Correct Answer ............................................................... 143
B.8 Second Multiple Choice Problem ...................................... 144
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.9</td>
<td>Wrong Answer</td>
<td>144</td>
</tr>
<tr>
<td>B.10</td>
<td>Hint</td>
<td>145</td>
</tr>
<tr>
<td>B.11</td>
<td>Choice Condition: Selecting a Problem</td>
<td>145</td>
</tr>
<tr>
<td>B.12</td>
<td>Choice Condition: Problem Selected</td>
<td>146</td>
</tr>
<tr>
<td>B.13</td>
<td>Choice Condition: Correct Answer</td>
<td>146</td>
</tr>
<tr>
<td>B.14</td>
<td>Test Intro</td>
<td>147</td>
</tr>
<tr>
<td>B.15</td>
<td>Test Question 1</td>
<td>147</td>
</tr>
<tr>
<td>B.16</td>
<td>Test Question 2</td>
<td>148</td>
</tr>
<tr>
<td>B.17</td>
<td>Test Question 20</td>
<td>149</td>
</tr>
<tr>
<td>B.18</td>
<td>Test Finished, Continue</td>
<td>149</td>
</tr>
<tr>
<td>B.19</td>
<td>Test Finished, Achieved Mastery!</td>
<td>151</td>
</tr>
<tr>
<td>B.20</td>
<td>Bye</td>
<td>151</td>
</tr>
<tr>
<td>C.1</td>
<td>CITICompletionReport</td>
<td>153</td>
</tr>
<tr>
<td>C.2</td>
<td>IRB Approval 2009</td>
<td>154</td>
</tr>
<tr>
<td>C.3</td>
<td>Consent Form 2009, Page 1</td>
<td>155</td>
</tr>
<tr>
<td>C.4</td>
<td>Consent Form 2009, Page 2</td>
<td>156</td>
</tr>
<tr>
<td>C.5</td>
<td>Consent Form 2009, Page 3</td>
<td>157</td>
</tr>
<tr>
<td>C.6</td>
<td>Consent Form 2009, Page 4</td>
<td>158</td>
</tr>
<tr>
<td>C.7</td>
<td>IRB Approval 2010</td>
<td>159</td>
</tr>
<tr>
<td>C.8</td>
<td>IRB Amendment 2010</td>
<td>160</td>
</tr>
<tr>
<td>C.9</td>
<td>Consent Form 2010, Page 1</td>
<td>161</td>
</tr>
<tr>
<td>C.10</td>
<td>Consent Form 2010, Page 2</td>
<td>162</td>
</tr>
<tr>
<td>C.11</td>
<td>Consent Form 2010, Page 3</td>
<td>163</td>
</tr>
<tr>
<td>C.12</td>
<td>Consent Form 2010, Page 4</td>
<td>164</td>
</tr>
</tbody>
</table>
LIST OF TABLES

4.1 Finite Field Binary Expression Tables ........................................ 54
4.2 Finite Field Arithmetic Problems To Teach ................................. 56
4.3 LAFF Training Problem Set ...................................................... 57
4.4 Pre/Post Test for LAFF Experiments .......................................... 58
4.5 Expert Problem Presentation Order ............................................ 59
4.6 Experiment I: Subject Counts ................................................... 66
4.7 Experiment II: Subject Counts .................................................. 68
4.8 Experiment I: Mean Test Scores (with standard error) ................. 69
4.9 Experiment I: Mean Test Scores on Trivial Problems (with standard error) ................................................. 70
4.10 Experiments I & II: Mean Test Scores on Trivial Problems (with standard error) ................................................. 71
4.11 Experiments I & II: Mean Test Scores (with standard error) .......... 72

5.1 Definitions: $N$ and $C$ ............................................................ 84
5.2 Definitions: $N$ and $C$ ............................................................ 86
5.3 Expert Skill Presentation Order .................................................. 88
5.4 Expert State Space Example ....................................................... 89
5.5 Number of Subjects Per Condition and Number that Tested Out Early105
5.6 Mean Test Score (proportion correct) ........................................... 108
5.7 Mean Improvement ...................................................................... 112
5.8 Mean Improvement by Choice ...................................................... 112
5.9 Mean of First, Last, and Improvement ......................................... 113
5.10 Mean Number of Learning Events Before Test 0 ........................... 114
5.11 Mean Test Score Improvement By Gender .................................... 115
5.12 Choice vs No Choice: Mean Test Score and Improvement By Gender 115
5.13 Improvement from the first test to the last by Condition, Gender, and Cluster. The average values for tendency toward problems of types $M$, $W$ and $F$ are also shown. ................................................. 118
Throughout history education has been restricted to a relatively small percentage of the world’s population. The cause can be attributed to a number of factors; however, it has been chiefly due to excessive cost. As we enter the information age it becomes conceivable to make education freely available to anyone, anywhere, anytime. The Intelligent Tutoring System is an automated teaching system designed to improve through experience, eventually learning to tailor its teaching to perfectly match each individual student’s needs and preferences. In this dissertation we describe a template which we use for building problem-oriented skill teaching intelligent tutoring systems based on a Dynamic Bayes network framework. We present two case studies in which the template is adapted to very different teaching domains, documenting in each case the process of building, training, and testing the resulting ITS. In both case studies, the performance of the ITS is validated through human subject experiments. The results of these studies show that our template is a viable technique for designing ITSs that teach in skill based domains. We also show that, while conducting artificial intelligence research on the design of an ITS and collecting data for use in that regard, we can concurrently run educational research experiments. We find that the two are quite inextricably tied and that showing good general results regarding the performance of the ITS is not sufficient; a strong understanding of the experience of the students is also required. We report some interesting results covering the effect of choice in learning and a gender bias that shows up in our tutoring system.
CHAPTER 1

INTRODUCTION

1.1 The Availability of Education

Throughout history education has been restricted to a relatively small percentage of the world’s population. The cause can be attributed to a number of factors; however, it has been chiefly due to excessive cost. As we enter the information age it becomes conceivable to make education freely available to anyone, anywhere, anytime. Personal computing devices and internet connections are becoming more and more commonplace around the world, bringing with them access to a multitude of educational resources. These resources include a number of exceptional automated and semi-automated teaching tools. For example, the Kahn Academy (http://www.khanacademy.org) is an online school with free enrollment which offers classes in a wide variety of subjects including mathematics lessons from arithmetic through calculus and beyond! Systems such as this offer numerous advantages over traditional classroom style teaching. Students are able to choose their own time and location for each study session. They can work at their own pace and review whenever the need arises without any fear of being stigmatized by faster moving peers. Their choices of material to study are no longer limited to what is offered in the nearest school, nor is the quality of instruction dependent on the local set of teachers. As automated teaching systems become more sophisticated they approach an ideal where a system can improve itself through experience, eventually learning to tailor its teaching to perfectly match each individual student’s needs and preferences.

The attainment of such a goal still requires a number of advances. Currently, the design and development of automated teaching systems is quite expensive with each system requiring a great deal of attention to domain-specific features. While a
human teacher may be able to teach a number of different subjects and can easily learn to teach new subjects, the typical automated system has a very specific domain and is entirely incapable of teaching anything outside of its focus. The development of a generalized automated teacher, able to teach effectively in multiple domains, adaptable to new domains, and capable of improving itself through experience, will represent a significant advancement.

In order to construct such a general teaching system we need to identify factors that are common to teaching and learning across domains and design a framework that is based on these commonalities yet is also able to learn about specific domains through experience. We would prefer that our system be capable of making use of (but not be dependent on) knowledge available from expert teachers, rather than taking a tabula rasa approach which would likely be undesirable for the first generation of learners.

In this dissertation we explore the possibility of constructing a generalized teaching system. We make progress toward identifying a set of factors required for effective teaching in any domain. The result of this work is a generalized framework which, while not immediately adaptable to any new task domain, is relatively easy to port to new domains with little change in the basic structure. We present two case studies in which we use the same underlying framework to teach different domains. For each example domain, we explain how we use data collected from students to allow our system to learn its own teaching policy for each domain. We then report empirical results on the performance of students trained under various handcrafted policies as compared to students trained by a teaching policy automatically constructed by our system.

1.2 The Role of a Teacher

1.2.1 A Domain Expert

In order to excel at teaching, a teacher must have a strong understanding of the subject matter. The main role of a teacher is that of organizing and presenting a
collection of information in a manner designed to ease another individual’s attempt to learn the material. Effective teachers do not present information willy-nilly without regard to the content. Instead, they seek to find any structure within the information that can be exploited to aid a learner’s understanding and retention of the material. Any interesting subject has some such underlying structure and any subject that does not is unlikely to require a teacher.

In general, an individual with mastery of a subject can be characterized as having a set of skills covering the important aspects of that subject. Typically skills need to be acquired in a strict or partial order since some skills will depend on others. For example, if we consider arithmetic to be a skill, it is unlikely that a student can gain mastery of algebra without first having mastered arithmetic. The algebra skill depends upon prior mastery of the arithmetic skill. However, within arithmetic, the order of acquisition of the skills of addition and subtraction may not be important\(^1\), either can be learned first. Teachers make use of these inherent skill hierarchies by breaking subjects down into subparts and ensuring that students have gained mastery of any pre-requisite skills before introducing new material. We see this process reflected at many levels in education including the division of schools into grades, the pre-requisite requirements of college courses, and the organization of material within individual courses. It should be possible to take advantage of this similarity in structure across domains to help build a domain-general teaching architecture.

1.2.2 Knowing the Student

Although subjects in different domains will have inherent orderings due to the dependencies between their skills, we cannot assume that by defining a partial order over the skills to be learned we will have discovered the teaching paragon for that domain. If students were clones, or automata, this would perhaps suffice; but given the complexity of the human brain things are not so simple. The different preferences

\(^1\)This statement is made simply to illustrate the notion of a partial order among skills, no argument is being made about the proper order of lessons in arithmetic.
and requirements of individual students add greatly to the difficulty of the teacher’s task. Teachers must know how to assess student proficiency levels at different skills and how to track student progress. They must also be able to recognize a variety of other factors about students including such things as affective state, alertness, and interest levels. Together these abilities allow teachers to identify specific individual needs of students.

Good teachers have a large repertoire of teaching methods and knowledge about when and how to use them most effectively. Throughout their training and careers, they learn to distinguish how different students respond to the variety of teaching techniques they are familiar with and what techniques work best under different conditions. In effect, they learn to empathize with different students, imagining themselves in the state that the student is in, enabling them to predict the effects of different actions and select the most appropriate. A teaching system will need a way to perform similar functions.

1.2.3 The Teacher’s Task

A teacher’s purpose is to ensure that students make optimal progress toward mastery of the subject being studied. In order to do so they must make decisions based on their knowledge of the domain, their understanding of the state of their students, and their expertise in teaching methods. Teachers are greatly hindered in their decision making by the fact that they typically teach many students together. It is not possible to teach to a group and at the same time address all individual needs. In particular, students have different time requirements for learning (Gettinger, 1984) indicating that the pace of classroom work cannot match all students needs. When a teacher decides to move on to a new topic, any student who has not yet mastered the current set of skills may experience difficulties learning the next topic. This cycle builds on itself remorselessly. As suggested in Bloom (1968), such gaps in understanding will cause a far-reaching cascade of problems for a student. We see the results in the plethora of remedial courses that must be offered at the college level in subjects that should have been learned during mandatory high school attendance.
Not only do teachers have to consider the pace at which they introduce new material, but also, they must order material in the best way. In some instances this may be obvious, as in the case of presenting arithmetic before calculus. On a more local scale, for example at the level of individual math problems, it is not always a simple matter to decide the best order of presentation. Through experience, teachers may learn that certain types of problems cause confusion when introduced too early and that other problems tend to be more helpful in the initial stages of learning. At any point in the teaching process a teacher must select the best action to perform (e.g. pose a math problem, give a hint, repeat a lesson) and the actions selected have interdependencies. Since learning builds on learning the decision of what to teach next is of critical importance and a teacher must reason about a student’s future in order to make the best decision in the present.

An automated intelligent teaching system specifically designed to perform as a one-on-one teacher, able to teach a variety of subjects, and able to learn through experience would greatly improve the educational experience for teachers and students. The availability of such an individual tutor for every student at any time would certainly alleviate much of the difficulty that students face in the classroom setting.

1.3 Intelligent Tutoring Systems

There are many different types of automated teaching systems and terminology varies widely. Enumerating all of the varieties is beyond the scope of this work and we refer the interested reader to Modritscher et al. (2004) for an attempt to classify them all. Our research focuses on the individual learner and his interactions with a teacher, leading us to adopt the Intelligent Tutoring System (q.v. Corbett, Koedinger, and Anderson, 1997) as the most appropriate form for our purposes.

An Intelligent Tutoring System (ITS) is an automated teaching system that uses Artificial Intelligence (AI) to specifically address the need for an adaptive individualized education. The ITS performs the role of a teacher, determining its best
course of action at any time by consulting a set of models covering factors vital to a successful interaction between a teacher and a learner. The quality of instruction depends on the accuracy of the system’s models. The typical ITS (Shute and Psotka, 1996) requires at least three main models referred to as the expert, the student, and the tutor. These three requirements for an intelligent teaching system, outlined as early as 1973 by Hartley and Sleeman, correspond to a representation of an expert in the domain, a representation of the student being taught, and a set of teaching operations with rules about their use.

1.3.1 Domain Expertise

The expert model is a representation of the subject matter to be taught structured for the use of the ITS. If the ITS teaches Spanish, for example, the expert may have information about verb conjugations, sentence construction, vocabulary, etc. If it teaches algebra it may contain important formulas, solving techniques, information about common misconceptions, and so on. The expert gives the tutoring system access to any domain knowledge a teacher might need. In general, this covers standard teaching materials (e.g. problems, readings, study guides, hints, videos, etc.) and performance testing tools. It may also include any knowledge about teaching and learning specific to the domain (e.g. what topics are best learned first, what proficiency level should be acquired in various sections before moving on, what areas typically cause difficulties for students, etc.). Depending on the domain, the expert can be extremely complicated and therefore is usually time consuming and expensive to develop.

1.3.2 Modeling the Student

The student model tracks each human student’s state of understanding of the material being learned. It may also include information about affective state, interest level, alertness, and other such factors to help the system tailor each learning session to the changing state of the student. Unfortunately, almost all of the factors that
must be monitored are unobservable and therefore must be inferred. In addition to performance testing tools, human teachers may make use of a number of sensory cues to help them make complicated inferences about changes in student state. An ITS is limited by the medium through which the student interacts with a computer which most often is simply a mouse and keyboard. The typical ITS makes great simplifications in order to model the state of a student.

1.3.3 The Tutor’s Task

The tutor model implements teaching strategies and tactics taken from expert human teachers as well as incorporating AI algorithms to allow learning and adaptation (duBoulay and Luckin, 2001). It uses the expert and student models to plan the best curriculum for each individual. We refer to a single tutor-student interaction as a learning event and to a sequence of such interactions as a learning trajectory. The tutor’s goal is to design the optimal learning trajectory for each student by repeatedly choosing the best teaching action for that student given his state. This problem of making a succession of choices is called a sequential decision problem.

Together the three modules allow the ITS to perform one-on-one teaching adapted to the specific needs of each individual student. Ideally, such a system should equal or surpass the effectiveness of a human teacher and should be easily adaptable to new teaching domains.

1.4 Designing A Skill Teaching Intelligent Tutoring System

In this section we discuss some of the steps involved in designing an ITS, introducing some necessary background material along the way. We then describe the types of domains we will use for our experiments and discuss in general how we will train and test the system.

In order to build the models in our ITS, we need to identify the skills to be learned and the dependency structure between these skills. This process will likely be done with the aid of a domain expert. With this information we will construct
the framework for a statistical model of how students learn the different skills. We then train the model using data collected from students taught under a variety of hand-crafted teaching algorithms, including random presentation of the material. Roughly speaking, training the model amounts to calculating the possible effects each teaching action may have on a student. The tutoring system will use this model to determine the best possible sequence of actions to take for each student to most quickly bring the student to a state of skill mastery.

1.4.1 Defining Domain Skills And Skill Mastery

The first problem we examine is the enumeration of skills in a domain. This is not a simple matter. For a mathematical domain for example, it is not immediately clear whether arithmetic should be considered as a single skill, or should instead be broken down into addition, subtraction, multiplication and division. One might even wonder whether addition should be split into a number of component skills. We could, for example, decompose the addition skill into single digit addition and multiple digit addition. It may be useful to introduce a skill covering the notion that addition of zero to any number has no effect. Since the system will be reasoning about each student’s skill proficiencies in order to make decisions about what and how to teach him, the granularity of skill definition and how the skills interrelate is very important. If skills are too finely granulated the system can waste much time unnecessarily verifying a student’s understanding of many skills that would better be represented by a single subsuming skill. This could lead to inefficient performance of the system and loss of interest in students as they are continually required to prove their abilities at tasks that have become too easy for them. Introducing a large number of extraneous skills also causes an unnecessary increase in computational difficulties for the ITS. On the other hand, if skills are too coarsely defined it may become impossible to accurately pinpoint student deficiencies. Returning to the arithmetic example, a system that is only capable of noting poor student performance at the level of arithmetic may spend an inordinate amount of time presenting addition, subtraction, and division lessons to a student who is strictly having dif-
difficulty with multiplication. This again leads to inefficient system performance and increases in student frustration levels.

Having skill mastery as our primary goal requires a more formal definition of mastery and an effective means of measuring student mastery relating to each skill. Once we decide on an acceptable metric we must define a threshold or standard to compare students against. The definition of mastery is inseparable from this measurement and the mastery measurement will be used for at least two important tasks. Our system determines that a student is fully prepared to learn a new skill by ascertaining mastery of any prerequisite skills. Skill mastery also provides a way to directly compare students to each other, which in turn allows us to compare the effectiveness of different tutoring policies.

Since a student’s true understanding of a given skill is unobservable (it is in his mind), we must use some external measurement which will necessarily be an approximation. In order to estimate mastery we examine performance in different situations designed to test a student’s abilities related to a given skill or skills. Our representation of skill mastery will be a discrete value based on these tests. Here we have another granularity issue. We need a way to represent each student’s current level of mastery of each skill but it is not directly apparent how fine a gradation of mastery levels is required. It may suffice to simply consider skill mastery as a binary value, either true or false for each skill. Or, it may be the case that the system can gain valuable information from knowing that a student is “close” to mastery. The granularity of gradation we use will have an effect on the system’s ability to distinguish different states of knowledge of different students and will therefore have an effect on its decision making.

1.4.2 The State Space

Our choice of how to represent skills and skill mastery will also have a large effect on our ability to compute solutions. Suppose there are $n$ total skills to learn and we decide that dividing mastery into $m$ discrete levels will be sufficient for our purposes. If we consider each of the $n$ skills as a dimension in a Cartesian space
then each dimension has $m$ positions along it and there are $m^n$ possible states that each student can be located in. We call this a state space. Even if there are only 2 skill levels (mastered and not mastered), the size of the state space is already at risk of intractability in any domain with more than a handful of skills. This assumes that all we are interested in about a given student is his mastery level on each of the relevant skills at a single moment in time. Tracking more information about each student beyond their mastery levels will only increase this difficulty. Therefore, we are strongly inclined to reduce the number of factors tracked and to keep the granularity of representation of each factor as low as possible. We will achieve this state space reduction in several ways.

1.4.3 The Markov Assumption

We will make the simplifying assumption that student learning can be modeled as a Markov process. In a Markov process, history is irrelevant given the current state. More formally, the probability distribution over all possible next states is conditionally independent of past states given the current state. In the case of student learners this means that the ITS does not need to know the entire developmental sequence of each student in order to make intelligent decisions about how to best teach the student. This Markov assumption allows us to ignore a student’s history which greatly reduces the amount of information we must track about each student. All that is required is a good description of the student’s current state. We make the assumption not because we believe it to be a true statement about human learners, but rather because we believe that the benefits it grants with respect to the ITS will far outweigh any possible adverse effects. As an added benefit to the great reduction in the state space size, the Markov assumption allows us to apply well known algorithms toward solving our sequential decision problem.

To model each student’s state of knowledge we maintain a vector with entries corresponding to each of the skill proficiencies students can acquire. Modeling student progress as a Markov process allows us to model the effect of actions in our tutoring domain as a Markov Decision Process (Puterman, 1994) or MDP defined as a set
$M = \langle S, A, R, T, \gamma \rangle$ where $S = \{s_1, s_2, ..., s_n\}$ is a set of states the environment can be in, $A = \{a_1, a_2, ..., a_t\}$ is a set of actions that can be taken, $T(s, a, s') \rightarrow Pr[s']$ is a transition function giving the probability that action $a$ taken in state $s$ will result in state $s'$, $R(s, a, s') \rightarrow \mathbb{R}$ is a reward function, and $\gamma$ is a discount factor.

In our case the set of states will be all possible states of student knowledge under the set of skills and mastery levels defined for a given domain. The set of actions are defined by the set of all possible tutoring actions for the domain. The reward function will be based on the number of skills mastered implying a preference for mastering as many skills as possible. The transition function will give us the probability of a student making a transition from one state of knowledge to another due to a particular tutoring action. The discount factor will serve to encourage the tutor to bring the student to mastery quickly. Once populated with data, this structure is a statistical model of how student knowledge changes due to different teacher actions.

Accurately predicting the effects of different teaching actions is essential to the tutor’s task of sequencing learning events for a student. To make such predictions, we will need to collect and analyze data showing the actual effects of different actions on human students. The MDP’s transition function depends on this information. Once the transition function is fully defined, an MDP can be solved mathematically to give an optimal policy (e.g. a best tutor action choice from any student state) which we can use as a solution to our sequential decision problem (see Kaelbling, Littman, and Moore (1996) and Russell and Norvig (2003) for surveys of reinforcement learning including MDP’s and solving techniques).

We still suffer with difficulties stemming from the size of the state space. As the size of the transition table grows, the computation time taken to solve the MDP increases. Since the size of a tabular representation of the transition function is exponential in the number of factors it quickly exceeds any reasonable amount of computation time. In addition to the problem of computation time, we face a data collection difficulty. Assuming our tutor has a set of $t$ teaching actions available it needs the ability to predict the effect of each of those actions from any of the
$m^n$ states that students may be in. That is, in order to help a student make the proper transitions in state that will eventually lead him to mastery of all of the skills, the tutor needs to know how to select an action $a$ to perform based on the student’s current state $s$ that will cause the student to transition to a state $s'$ which is the most desirable next state from the tutor’s perspective. If states are composed of $n$ factors with $m$ possible values for each factor (in the general case each factor might have a different range of values) then there will be $m^n$ total states and the transition function will contain $m^2n$ cells (if each state can possibly have any other state as a next state). That is, the size of the table will be exponentially large in $n$. In a deterministic world every action always has the same effect when performed in a particular state and we would only need to observe one instance of applying each action in each possible state to know its result. Human learning is obviously non-deterministic. Thus, in the worst case, we would need to collect multiple observations of the effect of each action taken in each state to have any confidence in our system’s ability to later predict the result of an action on an individual’s state. Collecting sufficient data to fill the size $t*m^2n$ transition table would be impossible given a finite, and likely quite small, set of human subjects. The Dynamic Bayes Net, introduced next, offers a solution.

1.4.4 Dynamic Bayes Net (DBN)

Using a vector representation of student state (a flat MDP structure (Puterman, 1994)) causes the exponential state space size increase. To avoid this problem we notice that in many instances an action will have no effect on many or most of the factors we are tracking. Each factor is conditionally independent of the majority of other factors given some small set of what we call its parent factors. From the skill dependency structure that we acquire from a domain expert, we can construct a factored MDP, modeling the transitions as a Dynamic Bayes Network (Dean and Kanazawa, 1989). In the Dynamic Bayes Net (DBN), each factor is only conditioned on a small number ($k = O(1)$) of parent factors effectively breaking the single large state transition function of the MDP into a set of small factor transition functions.
Figure 1.1 gives the intuitive idea of the difference. The drawing on the left side depicts a state transition under the flat MDP representation where factors are bound tightly together and the transition function maps a complete state description at time \( t \) to a complete state description at time \( t+1 \). This model is too specific and can say nothing about how each of the individual factors contributed to the transition.

The drawing on the right side shows a transition under the DBN representation. Here it is clear which factors influence other factors in a one step transition. Notice that while factor \( F_2 \) has no direct influence on factor \( F_5 \) it does affect \( F_3 \) which in turn has an influence on \( F_5 \). That is, the cascading effects of long range influences occur naturally in the DBN formulation.

The 2-layer structure depicted on the right side of Figure 1.1 shows each of the next state factors \( F'_i \) linked to its set of parent nodes \( \phi \) in the current time step \( t \). Each of the nodes at time \( t + 1 \) will have a corresponding conditional probability table (CPT) with a probability for each of its possible next values given a set of values of its parent nodes. Although we have represented the problem as a DBN, it is still Markovian with the full transition function for an MDP recoverable from the \( n \) factors in the DBN as: \( T(s, a, s') = \prod_{i=1}^{n} Pr[F'_i|\phi(F'_i), a] \). This is simply a product over the probability that each of the factors \( F'_i \) will take on the value in \( s' \) at time \( t + 1 \) if action \( a \) is taken, given that its parent nodes \( \phi(F'_i) \) had the values in \( s \) at time \( t \). Notice the number of parameters that need to be learned to specify the DBN representation is only \( O(n^k) \) which is far smaller than the fully enumerated state space. The factored structure of the DBN allows us to take full advantage of all data collected which reduces the amount of training data required.
1.4.5 Tutoring Domains For Research Purposes

Identifying a task domain for use in research experiments is a challenge in itself. We need a domain that is reduced in size from natural full scale classroom subjects to keep the experimental design and analysis tractable, however we need to ensure that the domains we use will capture the important aspects of full size domains to allow for our work to function well as we scale up. To this end, we reduce the teaching problem to a sequential decision problem. That is, given a set of skills to teach and a set of materials to teach them with, at any point in a student’s learning trajectory the ITS must select the most appropriate element to present to the student next. We will not address the problem of learning about the structure of the domain itself, but see Chapter 6 for a brief discussion of how structure learning might be performed. Instead, the system will be given information about the dependencies between the skills in the domain. A set of diagnostic actions, or rules about how to form such actions will also be supplied to allow the system to judge student skill mastery levels.

Our choice of the subject matter for learning domains on which to test our system reflects the fact that we are focusing on skill learning in general. We specifically select domains in which we can identify what we call “catalyst” skills, those skills that once mastered will aid in the learning of other skills within the same domain. To diminish individual variability among students and to reduce the confounding influence of different levels of prior knowledge we desire novel domains. To generally simplify the process of experimenting with human subjects we use domains with relatively few skills which can be learned within a one hour training session. This small domain requirement also helps ensure that we are able to identify and analyze general learning trends across students.

Under these domain constraints we find the best approach to be the design of artificial task domains. We design each of our domains to comprise a structured set of catalyst skills. If we find that students do learn the skills more effectively when the material is presented according to an intelligent policy than when presented
randomly, we can conclude that indeed there is some structure (not necessarily
the exact structure that we intended as we will discover) and that learning in the
domain does benefit from a better teacher. In this case we feel that, although not
strictly representative of typical subjects studied, these domains will indeed give us
insight into skill learning in general. If our system performs well in a variety of test
domains, we may apply the knowledge gained in these experiments in future work
on larger more realistic tutoring applications. Our system should be applicable to
any domain that has a hierarchy of skill dependencies, but is most easily adaptable
in cases where learning is done by solving problems.

1.4.6 Training The ITS

Although such a system should be capable of learning from the performance of
students taught by random, there is no reason not to make use of advice from
expert teachers. That is, there are many different teaching strategies that we can
be certain the system does not need to explore (for example teaching calculus before
arithmetic) and although it would certainly learn to avoid such strategies in the long
run, we can save a great deal of time by focusing on variants of more reasonable
strategies. This is analogous to a teacher taking the best teaching strategies from
other teachers and merging them into an improved strategy. We want to exploit
known strategies by borrowing from them but we do not want to mimic them. We
would also like to allow for exploration of unknown strategies in order for our system
to possibly improve upon known teaching strategies. For this reason we train our
learning algorithm using data collected from students taught by a variety of teaching
strategies, some expert, some random, some involving human choice. In effect, we
will have an automated tutor that can observe the performance of other tutoring
strategies, assimilating the best of what it sees into its own teaching policy.

In order to validate the performance of our ITS’s learned teaching policy we
require other policies for comparison. For a baseline teacher we will use a random
teaching policy in which teaching actions are selected at random from those avail-
able. We expect this policy to perform very poorly and any policy of value must
significantly surpass it. Since we are constructing artificial task domains no human expert teachers will be available for comparison. As an analogue to the human expert teacher we will hand-craft several “expert” teaching policies for each of our domains based on our knowledge of the structure purposefully built into the domain. At first glance, this suggests the question of how well our designed expert teaching policies might compare to a real human expert teacher. However, if students learning under our expert teaching policies perform significantly better than those trained by the random policy then we know the expert policies have some merit, and under the circumstances, they will in fact be the best known teaching policies for our domains. Our tutoring system will be trained using data collected from students taught by random and expert policies. In effect, our tutor learns from observing the performance of other tutors, some poor and some better. What we seek to show is that our tutor can learn to teach as well as or better than those teachers that provide its training data. Whether better teaching policies for a domain exist is not necessarily relevant. In the best case, our tutoring system would significantly surpass the expert policy, but performing at an equal level is certainly satisfactory.

Since our learned policy is trained using data from students taught by the expert policies we wish to compare it to, we must be wary of the possibility of our system simply mimicking the policies of the experts. For this reason we use a variety of expert policies as well as a random policy for training data. We will also do a qualitative comparison of policies during our performance analysis to verify that the learned policy has indeed done something new and interesting.

1.5 Focus Of This Dissertation

This dissertation has three areas of focus described in the following sections.

1.5.1 General Skill Teaching DBN Template For Curriculum Design

The major focus of this work is the design of an ITS framework that is capable of skill teaching, easily adaptable to different domains, and like a human teacher, learns
from experience. In Chapter 3 we describe a template for building a multi-layered Dynamic Bayes Net which models the dynamics of skill acquisition in problem-oriented Intelligent Tutoring Systems. To overcome some limitations of the 2-layer DBN, our template introduces a number of specialized internal nodes, but remains general enough to allow generation of DBN models for many tutoring domains. The parameters of the models are to be learned from prior student data and through planning provide a teaching policy for solving the sequential decision problem within the domain. We call this problem \textit{curriculum design} and provide evidence of the effectiveness of our end-to-end curriculum design system in human subject studies in two distinct domains. Since we are discussing the development of a framework that can be easily ported to new domains, we need to illustrate the types of modifications required for new domains and we must validate its performance in different domains. We apply our solution to two very different domains and describe the process in Chapters 4 and 5.

1.5.2 Complete End-to-End System Validation

In order to fully characterize and evaluate a tutoring system, we feel that there are a number of steps that must be described that parallel those steps involved in the development of a human teacher. A human teacher begins with training (from other teachers and through teaching experience). The teacher assimilates the information gained through training into his own teaching policy. Finally, the new teacher’s performance must be evaluated in the field by teaching his own students. In the ITS literature we often see reports documenting some of these steps, but there is a need for studies showing the full cycle from end to end. In our studies we report all steps of the process, showing that indeed our framework is useable and useful in systems that teach real material to real human learners. We collect data from human subjects trained by a variety of expert and non-expert policies. Using that data we train our ITS and generate a new tutoring policy. Most importantly, we verify the value of the learned policy by using it to teach a new set of human students. In this way we can actually verify that a system has learned from real human data
to perform well at teaching real humans. To the best of our knowledge, ours is
the first study showing the complete end-to-end development of an ITS, detailing
the construction of the system, collection of human subject data for training of the
system, and validation of the end product on human subjects.

1.5.3 Using The ITS As A Testbed For Educational Research

Since we are in the business of collecting data from human learners through a func-
tioning tutoring system, we would be remiss if we failed to take advantage of the
opportunity to perform some research regarding factors that affect human learning.
Using the end-to-end system described in the previous section we will not only be
able to evaluate the effectiveness of our learned teaching policies on human subjects,
we will also have a testbed for studying human learning under various conditions.

The Effect Of Choice On Learning

One of our research questions concerns the pedagogical value of choice. Rather than
strictly placing the selection of the next learning event in the hands of the tutor, we
are interested in whether students might benefit from gaining some amount of control
over that selection. The degree to which we allow these choices to alter a student’s
learning trajectory is also of interest. In Chapter 5 we introduce experimental
conditions to study this effect. In one condition we allow students to select the next
learning event from three events chosen by the tutor. This introduces a relatively
minor amount of control (comparable to allowing a student to select one of three
addition problems to tackle next). The second condition derives from Vygotsky
(1978) and his notion of a zone of proximal development (ZPD). As in the prior
condition, we offer the student three options; however, in this ZPD condition one
of the options exposes new material, beyond what the tutor has selected. In this
case, students will be able to modify the pace of introduction of new material, which
may allow them to spend more of their trajectory within their own zone of proximal
development. The results of this study are discussed in Chapter 5.
Gender Differences In Learning

We are interested in finding out how well our learned policies perform in general, but also whether they work equally well for all students. We attempt to retain approximately equal numbers of males and females in each experimental condition to allow for comparison of the two groups. In Chapter 5, Section 5.5.4 we discuss some interesting results concerning gender differences.

1.5.4 Thesis Statement

A domain general skill-teaching framework for Intelligent Tutoring Systems can be constructed that is easily adaptable to new domains and makes use of data collected from interactions with human students to learn effective and efficient teaching policies.

1.6 Chapter Overview

Chapter 2: RELATED WORK begins with a detailed description of Intelligent Tutoring Systems (ITS) expanding on Section 1.3: Intelligent Tutoring Systems. This is followed by a discussion of work that is specifically similar to our own in which we clarify the novel aspects of our work.

Chapter 3: GENERAL DBN TEMPLATE describes in detail the solution we give for the problem presented in Section 1.5.1 of this chapter. This includes descriptions of each of the specialized nodes we introduce to form the general DBN template.

Chapter 4: CASE STUDY: LAFF describes our pilot study in which we show how to apply our general DBN template to a novel mathematical domain.

In Chapter 5: CASE STUDY: BLAST we describe the core study in which we make use of the knowledge gained from the pilot study and apply our model to a more complex domain. In this chapter we also introduce a number of interesting gender related learning differences we discovered and give some insight into possible explanations for them.
Finally, we briefly summarize our findings in Chapter 6: CONCLUSIONS AND FUTURE WORK, suggest a number of future studies we would like to implement, and make some concluding remarks on ITS research.
In this chapter we first discuss educational research that strongly motivates the need for Intelligent Tutoring Systems (ITSs). This is followed by a general outline of the structure of Intelligent Tutoring Systems. We then give a description of research along lines similar to our own, clarifying the novel aspects of our work.

2.1 Human Education

An abundance of evidence exists (Gettinger, 1984) demonstrating that students differ in the amount of time needed to learn a given task. More importantly, research suggests that if each student is allowed to spend (and spends) the appropriate amount of time for that student, the majority of students can reach the same level of mastery. If we accept this research as valid, it is unreasonable to expect conventional schooling to be the optimal approach. When teachers are given a large heterogeneous group of students and a limited amount of time they are forced into a position where they must either address students collectively or distribute their time among the students. In either case many students will be receiving suboptimal schooling. Studies (e.g. Bloom, 1980) show that in the collective case teachers tend to focus their time on the top 5% of students in a class without even being aware of this themselves. The majority of students in a classroom do not gain the benefits of an interactive education. Since classes typically contain around 30 students, trying to address individual needs cannot possibly be effective for all students. In most cases reducing the size of classes is not an option.

Research such as Bloom (1973) suggests that students under one-to-one tutoring are on average approximately two standard deviations above those under conventional teaching conditions with respect to final performance metrics. In other words,
the average tutored student performed better than 98% of the students taught in
the conventional fashion! This result has lead to the adoption of the “2 Sigma
Problem” (Bloom, 1984) as a goal for many education researchers. Since providing
one-to-one tutoring for every student for every subject is obviously a far too expen-
sive solution, many researchers focus on whether we can modify our approach to
group instruction to make it as effective as or even better than one-to-one tutoring.
A number of different strategies have been developed and studied (see Learning for
Mastery in (Bloom, 1968; Block and Burns, 1976; Carroll, 1989) and Personalized
System of Instruction in (Kulik et al., 1979; Hartley and Sleeman, 1973), and many
other strategies (e.g. Brown, 2001)). Many of these show improvements over tra-
ditional teaching, however there are still problems with this approach. The bulk
of the burden is often placed on teachers but solutions involving extensive training
or retraining of teachers are less likely to be adopted. Since it is very difficult to
make overarching changes to the entire education system, much research is aimed
at devising strategies that are simple to incorporate into current curricula. This
of course severely limits possibilities. If not for the cost, it seems quite clear that
one-to-one tutoring has many advantages making it an ideal teaching solution.

2.2 Intelligent Tutoring Systems

Our purpose here is not to expound on all forms of ITS’s, but rather to outline the
structure of systems similar in flavor to our own. For a detailed overview of the vast
research on ITS’s see (Corbett et al., 1997; duBoulay and Luckin, 2001; Modritscher
et al., 2004; VanLehn, 2006) and for a review of the history of ITS’s see (Shute and
Psotka, 1996; Fletcher, 1999).

An Intelligent Tutoring System (ITS) is an automated teaching system that uses
Artificial Intelligence (AI) to specifically address the need for an adaptive individu-
alized education. Human teachers are hindered by the fact that they typically teach
large groups of students and must make decisions for the greater good. The typi-
cal ITS is designed to avoid this problem and perform optimally on an individual
level (but see Almond et al. (2009) for an exploration of methods for aggregating an ITS’s estimates of individual proficiencies into group estimates for a more teacher-centric view). Intelligent Teaching System might be a more appropriate name since it avoids the connotation of being strictly an assistant to some other educational scheme. An ITS is adaptable to both cases, it can be used as an educational supplement but also may be complete in itself. The ITS performs the role of a one-on-one teacher, determining its best course of action at any time by consulting a set of models covering factors needed for a successful interaction between a teacher and a learner. It uses AI algorithms to select, on the fly, the most appropriate next step for a particular student given the most up to date information about that student including his most recent interactions with the ITS as well as all data collected from other students in the past. The quality of its decisions depends on the accuracy of the system’s models.

2.2.1 Modeling

The typical ITS (as described in Shute and Psotka, 1996) requires at least three main models referred to as the expert, the student, and the tutor. These three requirements for an intelligent teaching system, outlined as early as 1973 by Hartley and Sleeman, correspond to a representation of an expert in the domain, a representation of the student being taught, and a set of teaching operations with rules about their use.

**Expert Model**

The expert component represents an expert on the subject matter to be taught. If the ITS teaches Spanish, for example, the expert will have information about verb conjugations, sentence construction, vocabulary words, and etc. If it teaches algebra it may contain important formulas, solving techniques, information about common misconceptions, and so on. The expert gives the tutoring system access to any such domain knowledge a teacher might need. In general, this information covers standard teaching materials (e.g. problems, readings, study guides, hints, videos,
etc.) and performance testing tools (practice problems, quizzes, tests, etc.). It may also include any knowledge about teaching and learning specific to the domain (e.g. what topics are best learned first, what proficiency level should be acquired in various sections before moving on, what areas typically cause difficulties for students, etc.). The expert model will define the full set of actions available to the tutoring system. Depending on the domain, the expert can be extremely complicated causing it to be time consuming and expensive to develop.

**Student Model and State Spaces**

In general, the student component is an attempt to model each human student’s state of understanding of the material being learned. It tracks each student’s performance allowing the ITS to make informed decisions about when to repeat old material and when to move on to something new. It models anything about a student that may help the system to make better decisions regarding the student’s education. The student model may include information about affective state, for example his engagement (Beal et al., 2007a; Beck, 2005), or motivation (Johns and Woolf, 2006), and even the curious “Yes!” moments of Muldner et al. (2010).

Research on student modeling takes on a number of forms, including Item Response Theory (e.g. Johns et al., 2006), MDPs Almond (2007b), Bayesian Knowledge Tracing (Baker et al., 2008), Hidden Markov Models (Beal et al., 2007a), and an interesting web based approach (McCalla, 2004). Drigas et al. (2009) gives a nice review of the wide variety of AI approaches to student modeling from 1999 to 2009.

**Tutor Model**

The tutor model uses the expert and student models to plan the best curriculum for each individual. The tutoring component may include algorithms that allow it to improve its own performance as a tutor through experience. It also incorporates pedagogical knowledge from the field of education about how best to teach material of different types in different situations (duBoulay and Luckin, 2001). Some research
focuses specifically on verifying the effectiveness of pedagogical strategies (Chi et al., 2010).

2.2.2 The Goal Of The ITS

ITS’s differ greatly regarding what data they collect and how they make use of it. Systems are often strongly biased by the domain that they focus on, relying on heuristics and sets of rules that tie them to their domain. The goal of most ITSs is to provide individually adapted instruction in a very specialized domain. Some examples of different domains include a physics tutor (Conati et al., 2002), logic proof tutoring (Barnes et al., 2008), helping dementia patients complete daily tasks (Hoey et al., 2010), aiding welfare case workers (Dekhtyar et al., 2009), learning arithmetic and fractions (Beal et al., 2010), a tutor for lexical practice (Heilman et al., 2006), and a LISP programming tutor (Corbett and Anderson, 1995). Incidentally, Corbett and Anderson’s 1995 paper may also serve as an excellent choice to introduce students to ITS research as it presents a very thorough story and it introduces a number of important concepts including mastery learning, representing domain knowledge, modeling student knowledge and skill mastery, experimental evaluation of the system, and in-depth analysis of student performance and how to make use of the results. Chi and VanLehn (2007) reports a study on difficulty of porting an ITS across domains in an effort to reduce development costs. It is clear that a domain general solution would have great value.

Regardless of the domain, the ideal of any ITS is to teach as effectively as or better than a human tutor. A number of studies establish that ITSs indeed are capable of tutoring on par with human tutors (Green et al., 2011b; Beal et al., 2007b). Some (Corbett, 2001) report even more positive results.

2.3 ITS Work Related To Our Own

A number of features distinguish our work from prior efforts in intelligent tutoring. These include the setting (curriculum design rather than tutoring for a single prob-
lem type), the use of a multi-layered DBN with learned parameters and treating the problem as a sequential decision making planning task. There are, however, many related works in ITS that share traits with our approach. Many traditional ITS systems that follow students as they work through a specific type of multi-step problem (e.g. solving a physics problem) utilize Bayes Nets in some capacity, usually to model the partially hidden state of the student. A nice overview of modeling students using Bayesian networks is given in Millán et al. (2010). An example of this sort of modeling is seen in the physics tutoring system presented in Conati et al. (2002) which uses Bayes Nets (but not MDP-style DBNs) to keep track of the probabilities that a student knows certain facts and skills and uses a rule-based policy to give hints based on its belief state about the student. Other researchers have employed a data-driven approach (as we do in our learning system) to train such Bayes Nets from expert trajectories. For example, in an ITS system for teaching number factorization (Manske and Conati, 2005), input from domain experts is used to design Bayes Nets for tracking student state and updates to model parameters are learned during student interactions. However, these works did not treat the problem as a traditional planning problem, instead they a fixed rule-based policy and focus on the Bayes Net’s accuracy at tracking student state. Unlike these approaches that track a student’s state and map this to a hand-coded policy, we will be using our learned DBN model to create the teaching policy itself. Other approaches have considered skill acquisition from a Markovian perspective. Work on tutoring students for completing logic proofs (Barnes et al., 2008) used an MDP to model the state of the student and planned out a policy for when to give hints. However, that work was again focused on a single type of multi-step problem and used a tabular representation of the MDP (not a DBN).

The closest work to our own is the approach of Almond (2007a), who used a multi-level DBN incorporating the “bowtie” Net idea adapted from Mathias et al. (2006, 2007) to model the skill acquisition process. However, instead of using simple proficiency rules as we did, that work considered skill proficiencies to be partially observable and used a particle filtering method to link the observations and underly-
ing states. Although planning was discussed in that work, the DBN parameters and policy were hand-made and were used for state tracking with a hand coded policy and the only experiment was run on an artificial student. By contrast, we show how the CPTs of a differently structured DBN can be learned from human derived data and planned with to create a policy that we then verify the effectiveness of by testing it on human subjects. Approaches in other areas have used architectures with multi-level learned DBNs and planning. Examples include using learned DBNs to suggest actions to a welfare case worker Dekhtyar et al. (2009) and helping dementia patients complete daily tasks Hoey et al. (2010).

In summary, what we find to be lacking in the literature is the design of a domain-general ITS that that captures the notion of skill teaching and is easy to port between domains accompanied by studies specifically showing the full end-to-end process of collecting data from human students, learning a policy from the data, and evaluating its effectiveness by comparing the performance of students trained under the learned policy with students trained under policies that are known to work. In the next chapter we present a multi-layer DBN structure we use for building skill-teaching ITSs in different domains. We follow with two case studies detailing the effectiveness of our system.
This chapter describes a template\(^1\) for designing an Intelligent Tutoring System (ITS) that sequences learning materials in the most effective order to teach a student to quickly master multiple skills. We call this problem curriculum design and an ITS based on our template will be effective for any subject matter which can be broken down by a domain expert into subtopics between which there are inherent ordering dependencies (i.e. any learning scenario in which learning builds upon itself.) A resulting ITS is appropriate for individual use either online or in a classroom setting and can be used to complement regular classroom work, or as a stand-alone tutorial system.

In the first section we present some requirements on the types of domains for which the template will be applicable. Section 3.2 describes the specialized nodes introduced in our DBN and the resulting structure. In the final section, we discuss learning and planning within the model to produce the teaching policy.

### 3.1 Domain Requirements

#### 3.1.1 Model Parameters

The domain to be taught will have a set of skills \(\mathcal{S}\) and each skill \(s \in \mathcal{S}\) will have a set of dependencies \(\mathcal{D}(s)\) whose members are the immediate “pre-requisite” skills of \(s\). For example, in a mathematical domain the skill of multiplication might have addition as a pre-requisite. In this work, we assume that \(\mathcal{D}\) maps each skill only to the skills that it directly depends on, and that the dependency graph is acyclic. For instance, in our mathematical example, if understanding exponents depends on

---

\(^{1}\)Material in this chapter is based on work published in Green et al. (2011a,b).
having the multiplication skill we do not include addition in the dependencies of the exponent skill.

We make a distinction between skills and facts (similar to that between procedural knowledge and declarative knowledge, (e.g. Anderson et al., 1996)) defining a set of facts $F$ which may be needed for answering certain problems or for understanding hints. For example, in a language learning domain where skills involve using a given set of words to construct sentences that describe a picture, we consider the meanings of the words to be facts. That is, the student’s vocabulary would be considered as a set of facts while his ability to conjugate verbs or to arrange words into a meaningful sentence would depend on a variety of skills.

The domain will also define a set of teaching actions $A$ appropriate for teaching the skills where $A(s)$ is the set of all actions $a \in A$ that can be used to teach $s$ (note that $A$ can be defined algorithmically and therefore may be infinite in size). In the domains that we consider in our case studies (presented in Chapters 4 and 5) we further divide $A$ into two types. The actions available to the teacher are to either present a hint about a skill $H(s)$, or to ask the student to solve a problem about the skill $P(s)$. Together $H$ and $P$ make up the complete set of all possible teacher actions in the domain. Teachers are not able to choose the exact problem or hint to present, they simply select the skill to teach and whether it should be presented as a hint or a problem. The actual grounded instance of the action is generated randomly from all available actions (hints or problems) about $s$. There are two reasons for this separation of the action from the grounded instances. First, since our goal is for the tutor to design a general curriculum (i.e. a policy) for how to best organize the teaching of skills it should not rely on specific problems. Secondly, as mentioned above, the set of actions for a given skill may be of infinite size. However, recall the granularity discussion from Section 1.4.1 in Chapter 1 and note that we can redefine our skills to gain more specificity if need be. It is unlikely that we will have the need to select specific instances of problems to teach a skill.

We refer to an individual interaction between the tutor and student as a learning event. In our example domains, a single learning event involves either the presenta-
tion of a hint for the student to contemplate as long as he likes, or the presentation of a problem to be answered by the student. From a history of learning events, called a trajectory, the system will model and track the student’s proficiency state \( \rho \) consisting of his proficiency at each skill, \( \rho(s) \). In our case studies, we take a simplified approach to tracking student state, using a set of proficiency adjustment rules that specify how to update \( \rho(s) \) based on the history of learning events. We store proficiency values as integers and in general our proficiency adjustment rules increment or decrement \( \rho(s) \) depending on whether the student answers problems correctly or incorrectly. Each skill has a maximum proficiency value and the goal of the tutor is to have the student reach the highest proficiency value possible \( \rho_{\text{max}} \) on each skill in as few learning events as possible. It should be purely a mechanical problem to substitute other more accurate, albeit more complicated student modeling methods (e.g. Corbett and Anderson (1995), Baker et al. (2008)).

In summary, our domain requirements include the following parameters: a set of skills \( \mathcal{S} \), skill dependencies \( \mathcal{D} \), facts \( \mathcal{F} \), and actions \( \mathcal{A} \) which may be further divided into action types (e.g. hints \( \mathcal{H} \) and problems \( \mathcal{P} \)).

### 3.1.2 Training Data

The DBN architecture will be specified using the node types defined in the following section, but the values for the individual CPTs will be mined from a set of trajectory data collected from students trained by other policies. To encourage exploration we include a random presentation policy and policies that allow user choice. To exploit existing expert knowledge about teaching we use various expert hand-crafted policies. In our studies, we trained the CPTs on data from a series of hand-crafted “expert” policies, random policies, and some policies where users choose the problems or hints. The mix of these data streams is important because mining data collected from a single specific deterministic policy would lead to simple mimicry of that policy. The use of human-guided exploration in the “choice” policies (where students could select from a set of problems) also yields information in many new areas that might lead to better policies. The DBN generalizes from the collection of
data, considering specific CPT contexts (based on only a few parents and possibly using aggregation nodes to generalize), which can result in a unique policy that is not just a simple combination of the data-collection policies. With training data collection limited by the number of human subjects we can attract, there will likely be “holes” in the CPTs where we have no evidence about whether or not a problem will be answered correctly in a given context. We will fill in the holes as “no change in state” to keep the agent in known areas of the state space. This discouragement of autonomous exploration is necessary since we are collecting data from actual human subject trials and many (overly optimistic) exploratory policies are likely to be unhelpful or even detrimental (Kearns and Koller (1999)). Exploration issues are discussed further in the conclusion.

3.2 DBN Structure

Our DBN representation contains a factor for every $s$ in $S$ and has a dependency structure based on $D$. Unfortunately, a 2-layer DBN will not capture several important aspects of a skill teaching domain that we would like to take advantage of. For example, the action $P(s)$ will likely have an effect on $\rho(s)$ but may have no effect at all on other skills. We can reduce the size of CPTs if we capture this relationship. More importantly, a 2-layer DBN would have difficulty in capturing correlations in changing proficiency levels. For instance, suppose a problem $P(s)$ influenced both proficiencies $\rho(s)$ and $\rho(s')$ and also some of the facts in the problem. Whether these proficiency levels go up or down depends on whether the problem is answered correctly or not and what specific facts show up in the problem, so there are a number of correlated variable effects that will be difficult to handle without computing some “intermediate” values for the next state after the ground problem is selected and the question is answered. We will resolve these and several other issues by introducing a more complex multi-level DBN structure (extending previous work with “bowtie” nets Dekhtyar et al. (2009)). The structure will remain general enough that DBNs for many domains could be generated from the given domain parameters.
3.2.1 On Representing General Skills

Intuitively, the skill dependency structure we find in a learning domain will appear as a partial ordering where each skill $s$ has a set of parent skills $D(s)$ each of which has its own set of parent skills and so on up to the root level set of skills. There are cases where this is a static interpretation of the DBN is too brittle of a structure. Rather than interpreting connections between nodes as dependencies on other skills, we should view the structure as showing how transitions from one proficiency state to another depend on various factors. Viewed in this way, we notice that for a particular $\rho(s)$, a transition to a higher or lower proficiency level depends on many factors beyond the list of pre-requisite skills $D(s)$. There are many details, for example, about the action that will affect transitions. Some actions will have no bearing at all on a given skill proficiency. The presentation of a subtraction problem will likely have no effect on a student’s proficiency at multiplying fractions. Other actions may only affect certain skill proficiencies under a specific set of conditions. For instance, in a language learning domain a student who does not know the meaning of 5 out of 6 words in a sentence will probably have no change in any proficiency from interacting with that sentence. However, when the same student is presented with another sentence identical in structure to the first in which he knows all 6 words, may learn some grammatical lesson from the sentence. We need to introduce specialized structures to the DBN to compactly represent complex features such as these. Generally, these structures keep the sizes of the CPTs small, model correlations in action outcomes, and keep track of the interactions between facts and skills.

3.2.2 Match Nodes

In a typical propositional DBN, factors are usually linked to the action node itself, and the possible actions are added as potential parent factor values in that node’s CPT. In our case this would add $2|S|$ possible parent values to each CPT because there are 2 actions per skill. However, most of these entries would be redundant,
because if the action is not $\mathcal{H}(s)$ or $\mathcal{P}(s)$ there is often no effect on $\rho(s)$. In Figure 3.1, we introduce a $Match$ node for each skill (blue nodes in the figures), that indicates whether the action is a hint for $s$, a problem on $s$, or not specifically about $s$. In this way, the CPT for $\rho(s)$ needs only to be increased by a factor of 3 rather than $2|S|$. 

3.2.3 Facts Known Nodes

While our goal in this work is to build policies for student skill acquisition, the ground problems we present also have facts in them. For instance, in the case study presented in Chapter 5 students must answer questions in an artificial language which requires not only proficiency at a number of grammatical skills, but also simply knowing the meanings of different words. Many domains will have sets of such facts that require nothing more than simple memorization, perhaps through repeated exposure. When evaluating a student’s proficiency at a given skill we do not wish to make the erroneous decision to decrease the proficiency rating when poor performance was due to an unknown fact rather than a skill deficiency. Thus we have a credit, or rather blame, assignment problem when assessing incorrect answers given by a student.

To solve this problem, we introduce $FactsKnown$ nodes, shown as red nodes
in Figure 3.2, for each type of action. When a ground action is presented, the FactsKnown nodes take on a binary value based on whether all the facts in the problem are known or not. These nodes are attached to the corresponding skill proficiencies to handle the credit assignment described above. Also, since some skills may be easier to learn than others when few facts are known the introduction of FactsKnown nodes allows for finer granularity in the CPT for advancing a skill.

In cases where keeping track of fact proficiencies is prohibitive, or during the planning phase when answers to specific questions will not be tracked, an estimate of the probability of knowing a fact based on the number of skills known can be used (assuming student progress and the number of learned facts are correlated). We were able to utilize this technique in the planning component of our artificial language case study. Figure 3.3 shows a plot of the number of facts known versus the number of skills known in our artificial language domain from which it is clear that students who knew fewer than three skills were very unlikely to know any given fact while those with more than ten skills were almost certain to know their facts.

3.2.4 Aggregate Nodes

Some nodes may depend on the number of mastered pre-requisite skills. There may be some threshold number of pre-requisite skills required before a student can gain
any information from a problem. If the pre-requisite structure can be specified in this way it will result in a significant decrease in the size of CPTs. For this purpose we include the *Aggregation* node, which counts the number of parent nodes with some value. For instance, the green nodes labeled “Agg” in Figure 3.4 count the number of parent skills for a $\rho(s)$ that have not reached the mastery level. This allows nodes that may depend only on the number of mastered pre-requisite skills to consider just this count, rather than all possible combinations, leading to an exponential decrease in the size of the corresponding CPTs. Note that these edges only encode potential dependencies between skills and likely do not on their own determine the optimal policy.

### 3.2.5 Cascade Links

Sometimes an incorrect answer from a student should lead to the update of multiple skill proficiencies in the same step and we must ensure that these changes are synchronized. For example, if a particular skill proficiency $\rho(s)$ is repeatedly decremented due to poor student performance on problem related to skill $s$, we must
eventually infer that he is lacking in pre-requisite skills in which case we should decrement his proficiency rating on the parents of $s$ to induce the system to train him further in those areas.

We enforce such correlations by having all the skills that might be affected by such a change linked directly (in the same layer) to the next proficiency factor for the skill $s$. We refer to these types of correlations, seen in the bottom layer of Figure 3.5, as cascade links.
3.2.6 Full DBN Template

Figure 3.6 shows the full set of specialized nodes and links together. In Chapter 4 we will see an example of the use of Match nodes and in Chapter 5 we see examples of all three node types as well as cascade links.

3.3 Curriculum Design Planning

3.3.1 Currently

Our core problem is to design a dynamic curriculum— a series of hints and problems that respond to a student’s proficiency state and push him to achieve mastery in the skills as fast as possible. Unlike other approaches that track a student’s state and map this to a hand-coded policy, we will be using our learned DBN model to create the teaching policy itself. We will do so by treating curriculum design as a planning problem with reward and transition functions specified with a DBN as described above. Viewed in this manner, we can calculate the value of each action \( a \) from a given proficiency state \( \rho \) as

\[
Q(\rho, a) = R(\rho) + \gamma \sum T(\rho, a, \rho') \max_{a'^{\prime}} Q(\rho', a'),
\]

and our policy is derived by choosing the maximum valued \( a \) at a given \( \rho \). Intuitively,
these values tell us what action will help us achieve full proficiency fastest, while also (through the discount factor) coveting skill proficiencies that can be learned sooner. This implicit trade-off between long-term teaching goals and short term skill proficiency is a key component of our approach and recognizes the practical limitation of short teaching sessions. Our purpose has been to study skill teaching and learning in a small, well-contained system, and in order to avoid complexity issues and stay focused on our main goals, we have purposefully kept our testing domains small. In our experiments we used Value Iteration (Puterman, 1994) to compute the $Q$-values and the corresponding policy, but this approach was at the limit of tractability (several hours).

3.3.2 Scaling Up

Traditional MDP planning algorithms like Value Iteration (Puterman, 1994) have a computational dependence on the size of the ground state space, which is exponentially large in the number of factors in a DBN. We were able to use Value Iteration in our current experiments, future iterations with larger state spaces will require approximate planners specifically designed for DBNs, such as SPUDD (Hoey et al., 1999). Approximate planners have shown their versatility in a number of real world problems (e.g. aiding dementia patients in household tasks, Hoey et al. (2010)). Also, recent research has shown that Monte Carlo Tree Search planners like UCT (Kocsis and Szepesvari, 2006) could be used to plan in portions of the state space without any dependence on the actual number of states. Advancements in computation power and planning algorithms make this DBN approach feasible for real world skill teaching scenarios, as described in our case studies.
CHAPTER 4

CASE STUDY: LAFF

In this chapter we describe our pilot study in which our ITS learned a policy for presenting problems in a novel mathematical domain. The ITS is compared to handcrafted tutoring policies by using each tutoring policy to train human subjects and analyzing their pre and post training performance.

In the first section we describe the mathematical domain and the structure of the tutoring environment. In Section 4.2 we describe a set of hand crafted tutors used for baseline and best performance comparison, and in Section 4.3 we describe the tutor learned by our system. Section 4.4 gives the details of the experiments and in Section 4.5 we provide the results and analyses. Finally, in Section 4.6, we briefly discuss the results mentioning some lessons learned from the experiments.

4.1 Domain Description: LAFF

4.1.1 LAFF: Learn Arithmetic in Finite Fields

In the pilot study, students learned to solve algebraic equations from an unfamiliar mathematical realm. Our subject population mainly consists of first year undergraduates who are unlikely to have had much, if any, exposure to abstract algebra. We take advantage of this bias in our population to reduce subject prior knowledge. We based our pilot study’s skill-acquisition domain on finite field arithmetic. We call the project LAFF for “Learn Arithmetic in Finite Fields” and students study and solve finite field arithmetic problems in order to learn how arithmetic works in finite fields.

To the uninitiated, arithmetic in a finite field is quite mysterious. Although the basic arithmetic operations are used, their behavior can be unexpected. Here we

\footnote{Some of the material in this chapter is has been published in Green et al. (2011b).}
use a field of size four\(^2\) with operands \(\{0, 1, A, B\}\) and the four standard binary operations \(\{+, -, \ast, /\}\). A mathematical expression is constructed in the expected fashion, e.g. \(A \ast B\). More complicated expressions can be constructed with or without parentheses. The order of operations is the same as in basic arithmetic, e.g.:

\[
A + 1 \ast B = A + (1 \ast B)
\]

Table 4.1 gives solution tables for binary expressions using each of the four operators and four operands. Reliably solving equations in LAFF does not require any deep understanding of finite fields and we provide no further explanation than what must be learned by the students in this study (for the interested reader, see Hankerson et al. (2004)). Rather, it requires practice solving different problems along with appropriate feedback. Once a student has learned the information in Table 4.1 and has verified that the order of operations is as expected (and that parentheses perform the normal function) it is possible to solve any equation of arbitrary length (providing that the student also understands the principal of reduction).

**FFA Problem Chains**

Initially, it appeared that this domain would nicely fit our requirements. It is relatively small and well constrained, allowing for more thorough analyses. It should be novel to most or all subjects which reduces confounds due to prior knowledge. It contains skills (e.g. multiplying \(1 \ast X = X\) for any \(X\)) that can be mastered. Once mastered, these skills can be used to solve more complicated expressions (e.g. \(1 \ast (X + 1) = X + 1\)). We predicted that students would easily learn the basic operations and be able to generalize those skills to solve more complicated expressions. After some preliminary trials however, it became clear that the skill dependency structure in the domain was not strong enough to elicit any interesting learning. Many (if not most) of the skills we had defined were learnable independently of the other skills (e.g. knowing addition in FFA does not appear to help a student to

\(^2\)that is a “Galois Field” of size 4 or \(GF(2^2)\)
Table 4.1: Finite Field Binary Expression Tables

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>−</th>
<th>0</th>
<th>1</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>B</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
<td>A</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*</th>
<th>0</th>
<th>1</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>0</td>
<td>A</td>
<td>B</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>0</td>
<td>B</td>
<td>1</td>
<td>A</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>/</th>
<th>0</th>
<th>1</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-</td>
<td>1</td>
<td>B</td>
<td>A</td>
</tr>
<tr>
<td>A</td>
<td>-</td>
<td>A</td>
<td>1</td>
<td>B</td>
</tr>
<tr>
<td>B</td>
<td>-</td>
<td>B</td>
<td>A</td>
<td>1</td>
</tr>
</tbody>
</table>

learn multiplication). Since we are not interested in a domain composed of mutually independent facts, which would probably be best served by rote memorization, we would like to ensure that skill learning in the domain is best served by teaching the various skills according to some partial ordering. That is, understanding some aspect of the domain facilitates the learning of some other aspect. In order for the pilot study to be useful, that structure needs to be clearly defined and understood by the researchers. Introducing a known order has further benefits. If there exists a known learning partial order then we know how an expert in the domain might proceed. We can compare performance of human subjects trained using an expert policy (which we can confidently derive from the domain) to subjects trained using a random policy to verify that teaching order is important and to find the improvement expected by using a good teaching policy. This gives us a nice baseline to compare performance under any learned policies against.

In order to impose the desired skill hierarchy we made a number of alterations to the original FFA domain. We restricted the set of FFA problems to 24 binary problems (i.e. one operator, two operands) taken from Table 4.1. We organized
these problems into 8 groups of 3 problems each. Within each group, problems were chosen such that the solution to the first problem appeared as an operand in the second problem and the solution to the second problem appeared as an operand in the third, for example:

1st) \( 1 + A = B \)

2nd) \( 1 - B = A \)

3rd) \( A \times A = B \)

We then expand the second and third problems by rewriting one operand as the preceding problem:

1st) \( 1 + A \)

2nd) \( 1 - B \Rightarrow 1 - (1 + A) \)

3rd) \( A \times A \Rightarrow (1 - B) \times A \)

If the problem \( 1 - B \) is never explicitly presented to the student then we can be sure that the only way a student can reliably learn the solution to \( 1 - B \) is to first know the answer to \( 1 + A \) and then see the problem \( 1 - (1 + A) \) along with its answer. This gives us the required skill hierarchy. We will refer to such a sequence of three problems as a “problem chain”, or simply a “chain”.

The set of problems used in training was designed to teach 22 problems taken from Table 4.1. Table 4.2 shows a list of the underlying problems that were taught. The 8 problems in the “Primary” column were presented exactly as shown Table 4.2. The other 16 problems were embedded inside of problem chains as described in the example above. The problems in row 2 were added to ensure that students had a chance to verify that symmetry works in this domain for addition and multiplication, bringing the total problem count to 24.

Table 4.3 shows the actual 24 problems presented to subjects. The problems in the “Primary” column were learnable as atomic units. The problems in the “Secondary” column require knowledge gained by learning problems in the “Primary”
<table>
<thead>
<tr>
<th>Chain</th>
<th>Primary</th>
<th>Secondary</th>
<th>Tertiary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1 + A</td>
<td>1 − B</td>
<td>A × A</td>
</tr>
<tr>
<td>2.</td>
<td>A + 1</td>
<td>1 − B</td>
<td>A × A</td>
</tr>
<tr>
<td>3.</td>
<td>1 + B</td>
<td>B ÷ A</td>
<td>A − 1</td>
</tr>
<tr>
<td>4.</td>
<td>A + B</td>
<td>1 ÷ A</td>
<td>B × B</td>
</tr>
<tr>
<td>5.</td>
<td>A × B</td>
<td>1 − A</td>
<td>A ÷ B</td>
</tr>
<tr>
<td>6.</td>
<td>1 + 1</td>
<td>0 − B</td>
<td>1 ÷ B</td>
</tr>
<tr>
<td>7.</td>
<td>A + A</td>
<td>0 − 1</td>
<td>B − 1</td>
</tr>
<tr>
<td>8.</td>
<td>B + B</td>
<td>0 − A</td>
<td>B − A</td>
</tr>
</tbody>
</table>

Table 4.2: Finite Field Arithmetic Problems To Teach

column. The problems in the “Tertiary” column require knowledge that gained by learning those problems in the “Secondary” column.

For example, row 1 in Table 4.2 shows that we would like to teach the subjects how to solve the problems: 1 + A, 1 − B, and A × A. Subjects will see an explicit example of 1 + A. However, they will never see an example of 1 − B, but rather, they will see 1 − (1 + A). If a subject has previously learned that the solution to 1 + A is B he should then be able to reduce 1 − (1 + A) to 1 − B allowing him to subsequently learn the solution to 1 − B. Likewise, a subject will not see the problem A × A but instead will see (1 − B) × A. If the subject has learned to solve 1 − B as A from the “Secondary” problem he should be able to reduce (1 − B) × A to A × A. The other rows are constructed similarly.

4.1.2 Teacher Actions

The only action allowed to a tutor (the various tutors used in the pilot study are described in Sections 4.2 and 4.3) was simply to select a problem from the set of 24 problems in Table 4.3 to present to the student next.
4.1.3 Evaluation

In order to measure the performance of the various tutors we designed the 15 question test shown in column 1 of Table 4.4 to be presented before and after training. The test is intended to broadly test FFA skills. All students took the same test before and after training. The purpose of the pre-test was of course to establish students’ level of prior knowledge. The purpose of the post-test was to analyze the differences in performance of students due to different training methods. The test contains problems of 5 types, shown in the second column. “Trivial” problems are those we assume that the students already know due to their pre-existing understanding of algebra, we include these to verify that students already knew these simple problems. “Primary”, “Secondary”, and “Tertiary” problems are problems that we expect to be new to students and that must be learn in the way outlined in Section 4.1.1. Finally, the “Symmetry/Composite” problems are designed to test whether the student has learned symmetry (commutativity) and to further test the students’ mastery of the core 24 problems. The last column lists the minimum set of training problems\(^3\) required for a student to learn to answer

\(^3\)training problems 8,11,14 do not appear in the table
Table 4.4: Pre/Post Test for LAFF Experiments

each test question\textsuperscript{4}. Unlike in training, where subjects received feedback on each problem in the form of the problem itself paired with the correct answer, the only feedback available for a test was the final score out of 15.

4.2 Handcrafted Tutors

Four handcrafted tutoring policies were used for comparison with the performance of learned policies.

4.2.1 Random Tutor (RND)

In order to establish a baseline, a random tutor was used in which the 24 problems were presented to each subject in a random order without replacement. For each problem $\frac{1}{6}$ & \{(0,1,2) or (3,4,5)\}
\begin{table}[h!]
\centering
\begin{tabular}{|c|c|c|}
\hline
\textbf{Test Problem} & \textbf{Type} & \textbf{Training Problems/Chains Needed} \\
\hline
(0 + A) ÷ A & Trivial & none \\
\hline
(1 × A) − 0 & Trivial & none \\
\hline
(A × 0) ÷ B & Trivial & none \\
\hline
B − (B ÷ 1) & Trivial & none \\
\hline
1 + 1 & Primary & 15 \\
\hline
A + B & Primary & 9 \\
\hline
A × (A + 1) & Primary & 3 & 12 \\
\hline
1 ÷ A & Secondary & (9,10) \\
\hline
B ÷ A & Secondary & (6,7) \\
\hline
B − 1 & Tertiary & (18,19,20) \\
\hline
B − A & Tertiary & (21,22,23) \\
\hline
B + A & Symmetry/Composite & \textsuperscript{5} \\
\hline
(B + 1) × (0 − A) & Symmetry/Composite & \textsuperscript{6} & (21,22) & \{(0,1,2) or (3,4,5)\} \\
\hline
(B × A) ÷ (1 − A) & Symmetry/Composite & \frac{1}{12} & (12,13) & (15,16,17) \\
\hline
(B + B) − (B − 1) & Symmetry/Composite & 21 & (18,19,20) & (21,22) \\
\hline
\end{tabular}
\caption{Pre/Post Test for LAFF Experiments}
\end{table}

\textsuperscript{4}the left arrow in the table marks problems that test knowledge of commutativity
subject RND presented a different random ordering of the 24 problems.

4.2.2 Expert Tutor (EXP)

An “Expert” tutor was designed to present the chains in the simplest order that would allow subjects to make the connections between links in each chain. Under the EXP tutoring, the 24 problems were presented to each subject in the fixed order shown in Table 4.5.

We assume that problem chains are most easily learned when each chain is presented in its natural order (primary, secondary, tertiary) before moving on to the next chain. The expert order is taken directly from Table 4.3, presenting each chain in its natural order. Each subject saw exactly 24 problems in the order given here.
4.2.3 Derived Tutor (DRV)

To verify whether a simple statistical analysis of the performance of subjects under random tutelage could provide enough information to create a passable tutor we designed a second fixed ordering tutor. Using the data collected from the subjects trained under the RND tutor, we attempted to extract information about dependencies between pairs of problems by looking at post-test scores for subjects who received problem pairs in different orders during training. Using the results we derived a new fixed ordering of the 24 problems to be used in the same way as the EXP tutor. In most cases the DRV order preserved the natural order of the chains, however it often broke the chains by injecting other problems between the chain links. As will be shown in the results section DRV performed very poorly and the idea was abandoned. We mention the DRV tutor since we included the data collected under DRV in the set of data used to trained the DBN. For complete details about the DRV tutor see Appendix A.

4.2.4 Finite State Machine Tutor (FSM)

A second expert tutoring policy was designed by hand in the form of the finite state machine (FSM) shown in Figure 4.1, in an attempt to simulate the behavior of a human teacher. The FSM policy grouped primary and secondary problems from the same chain in an effort to ensure mastery of the first two parts of each chain before moving on. The FSM was specifically tailored to teach to the test.

The FSM differs from the previous three tutors in two ways. The FSM allows repeated presentation of individual problems and its problem list includes several “Trivial” problems (which are introduced in Section 4.4.2) that were excluded from the three previous tutors.

4.3 Learned Tutor: DBN

The model for the learned tutor was based on the DBN template described in Chapter 3. The state factors used in designing the DBN tutor were simply the
Figure 4.1: Handcrafted FSM Policy
proficiencies at each of the 24 problems. Since we consider each of the primary, secondary, and tertiary problems as skills we have no need for Fact nodes and the simple parent structure of the problem chains does not require Aggregate nodes. The only internal node we require is the Match node. The top of Figure 4.2 shows the DBN structure used. In the top row of nodes, representing skill proficiencies at time \( t \), the first three nodes (\( P_0 \), \( S_0 \), and \( T_0 \)) represent the primary, secondary, and tertiary problems from the first chain. To conserve space we have removed chains 1-6 from the figure. Following the nodes for chain 7 there are two nodes labeled \( \text{Triv}_0 \) and \( \text{Triv}_1 \) to represent “trivial” problems (e.g. \( B - 0 \)) that were introduced in the second phase of the pilot study (see Section 4.4.2).

As described in Section 1.4.4 of Chapter 1, each of the nodes at time \( t + 1 \) will have a corresponding conditional probability table (CPT) containing the probability of making the transition to each of it’s possible next state values given a particular set of values of its parent nodes \( \phi \). An example what the CPT might look like for the \( S_0 \) node is shown at the bottom of Figure 4.2. The two rows highlighted in red show the case where the problem presented (i.e. the action) does not match the \( S_0 \) skill. The value of \( P_0 \) is irrelevant in all such cases, shown in the table as an “*” and the value of \( S_0 \) remains unchanged after the action with probability 1.0. The third
row indicates that without prior mastery of $P0$ an action-skill match still gives no chance for a transition from non-mastery to mastery of $S0$. The last two rows show cases where the primary problem $P0$ was already mastered and the action matched the $S0$ skill. The probabilities in the example are fabricated, but reflect the idea that mastery of $S0$ will likely be retained but gaining mastery is a less likely event.

The DBN was trained (i.e. the values in the CPTs were calculated) using data collected from subjects trained under the RND, EXP, and FSM conditions previously described in Section 4.2. None of these three tutors explicitly track a subject’s proficiency state. The RND and EXP conditions both essentially ignore the subject’s skill proficiency and simply continue to present problems according to their fixed rules. Proficiency state is implicitly tracked by the transition arcs of the FSM, but there is no explicitly tracked state value. Training the CPTs requires a record of each subject’s transitions between proficiency states defined in terms of the skills represented by the DBN. As explained in Section 3.1.1, we used a simplified approach to tracking subject state, employing a set of proficiency adjustment rules to specify how $\rho(s)$ (the proficiency level at a given skill $s$) is updated during a subject’s training. Accordingly, the training trajectory of each subject was recoded as a sequence of state transitions where $\rho(s)$ for each of the 10 skills represented in the DBN (i.e. the 8 chains and the 2 trivials) was either a 0 for an “unmastered” skill or a 1 for a “mastered” skill. For each subject, state was initialized to all 0’s and $\rho(s)$ was increased from 0 to 1 after a correct answer to a problem of skill $s$ and was decremented to 0 after a wrong answer.

Although these rules allowed the CPTs to capture the more granular changes occurring on a problem by problem level, the same rules when applied during training caused very abrupt changes in the tutor’s skill presentation. To alleviate this, the rules were modified to slow subject skill proficiency progression by requiring several correct answers to a given skill type before incrementing $\rho(s)$. During subject trials, $\rho(s)$ was increased by 1 (with a maximum of 2) after 3 consecutive correct answers to a problem of skill $s$ and was decremented by 1 after each wrong answer. We also assumed that secondary or tertiary problems could not be answered correctly
without having already mastered their dependencies (this deterministic effect of dependencies was relaxed in the case study presented in the next chapter). Thus, the main problem left to the DBN in this simplified domain was choosing what chain to pick the next problem from.

4.4 Experiment Details

4.4.1 Protocol

In this section we give an overview of the protocol used in the experiments, excluding details that are not relevant to the narrative. Complete protocol details can be found in Appendix A.

Each subject in the study received brief verbal instructions from the experimenter, on how the experiment would proceed. Each subject was then seated at a computer terminal and instructed to log in. The subject was given a printed copy of the diagram in Figure 4.3. Our intention was to focus on learning in the sense of understanding rather than solely on memorization skills. The purpose of this “cheat sheet” was to reduce variance among subjects due to memory differences and subjects were encouraged to “take notes on the sheet to help them remember what they had learned”. Logging in initiated the program and the experimenter returned to his own desk, in sight of the subject, leaving the subject to follow the instructions presented on screen. The experimenter did not interact with the subject again until the subject had finished the experiment.

Upon logging in, the subject was presented with a set of instructions listing the operators and operands used in LAFF and encouraging him to do his best. After reading the instructions, the subject was given the pre-test shown in Section 4.1.3 above. The same pre-test was given to all subjects in all conditions. Following the pre-test, the subject was presented with the sequence of 24 problems as prescribed by the tutor assigned to that particular subject. Problems were presented sequentially and the subject was allowed to view each problem as long as he pleased. Each problem was shown in multiple choice form with four options: 0, 1, A, B. After sub-
mitting an answer, the subject was presented with a brief congratulatory message, or a consolatory message suggesting that he pause to look over his notes. Following the training set, the subject was given a post-test (identical to the pre-test.) The same post-test was used in all conditions.

4.4.2 Experiments

In all experiments reported here, our subject pool was composed of university undergraduates, mostly between ages 18-20, all taking an introductory psychology course at the University of Arizona. Students received modest class credit for experiment participation, selecting the experiments they wished to participate in from a list of all Psychology Department studies in progress at the time.

**Experiment I: No Repetition**

The first experiment can be considered as a “calibration” of the LAFF domain. It focussed on assessing tutor performance (via subject performance metrics) under a tutoring task in which actions were rigidly restricted in order to reduce variance and allow for very thorough analysis. The task posed to the tutors was simply to present
the 24 problems in Table 4.3 in the best order to achieve the goal of teaching the skills required to solve FFA problems. That is, training for all subjects consisted of solving each of the 24 problems once and only once. Tutors were only allowed to manipulate the order of presentation of the 24 problems. The RND tutor was used to establish a baseline and the handcrafted EXP tutor was used to represent the performance of an expert human teacher. A third tutor, the DRV case described above, was an explorative idea (described in detail in Appendix A) in which data collected from the subjects trained by RND was analyzed to extract information about dependencies between pairs of problems which was then used to construct a problem presentation order.

This experiment adhered to the protocol described in Section 4.4.1.

A total of 73 subjects were available and the distribution among tutoring conditions is shown in Table 4.6.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Subject Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>28</td>
</tr>
<tr>
<td>EXP</td>
<td>23</td>
</tr>
<tr>
<td>DRV</td>
<td>22</td>
</tr>
</tbody>
</table>

Table 4.6: Experiment I: Subject Counts

**Experiment II: Repetition Allowed**

In the second experiment, the training requirements were relaxed slightly, again allowing a total of only 24 problem presentations per subject, but in this case letting the tutor repeat problems (and consequently skip other problems) if it chose to. In addition to allowing repetition, the problem set was augmented by the introduction of six trivial problems. Tutors were still restricted to presenting only 24 problems,

---

5Tutors were assigned at random for the RND and EXP conditions, but the DRV subjects were necessarily run after running all of the RND subjects.
but they had the following six additional problems to choose from (giving a total of 30 problems):

<table>
<thead>
<tr>
<th>$B \ast 1$</th>
<th>$B - 0$</th>
<th>$0 \ast 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$0/1$</td>
<td>$B + 0$</td>
<td>$B/B$</td>
</tr>
</tbody>
</table>

The 6 additional trivial problems are not problems seen on the test, but rather are representative of the types of trivial problem tested:

<table>
<thead>
<tr>
<th>Training Problems</th>
<th>Test Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B \ast 1$ and $B - 0$</td>
<td>$(1 \ast A) - 0$</td>
</tr>
<tr>
<td>$0 \ast 1$ and $0/1$</td>
<td>$(A \ast 0)/B$</td>
</tr>
<tr>
<td>$B + 0$ and $B/B$</td>
<td>$(0 + A)/A$</td>
</tr>
</tbody>
</table>

If, for example, subjects confirm by seeing $B \ast 1$ in training that $X \ast 1$ is in fact $X$, they should be able to simplify $(1 \ast A) - 0$ on the test to $A - 0$. If they are just recording answers to problems and looking them up during the test they should have more difficulty. During training subjects may also see $B - 0$ which allows them to confirm that $X - 0$ is indeed $X$ which should allow subjects to solve the simplified test question, $A - 0$.

The set of 6 trivial problems addresses the first three trivial problems on the test. The fourth trivial problem on the test is $B - (B/1)$ which is not represented in this set. Subjects get no direct confirmation that $X/1 = X$ nor that $X - X = 0$. If the 6 training problems are enough to convince subjects that problems that appear obvious by prior knowledge really are as they appear then they should be able to generalize and solve this problem.

Aside from these two changes (allowing repetition and introducing trivial problems) the second experiment adhered to the protocol described in Section 4.4.1.
A total of 53 subjects were available and the distribution\textsuperscript{6} among the tutors is given in Table 4.7.

<table>
<thead>
<tr>
<th>Condition</th>
<th>Subject Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSM</td>
<td>26</td>
</tr>
<tr>
<td>DBN</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 4.7: Experiment II: Subject Counts

After the entire FSM group had finished, the DBN was trained using the data collected under the FSM condition in addition to all data collected in Experiment I.

4.5 Results/Analysis

4.5.1 Experiment I

Experiment I compared the performance of three tutors: RND, EXP, and DRV. We first examine pre- and post-test scores. Given that the test had 15 multiple choice problems each with 4 options, we would expect a score of 3.75 by chance alone. However, 4 out of the 15 problems on the test were “trivial” problems which could be solved using prior knowledge of arithmetic. This leads us to expect even higher scores on the pre-test. In the best case, a subject might have a perfect performance on the 4 trivial problems and guess on all of the unknown problems resulting in an expected pre-test score of 6.75. Table 4.8 reports the pre-test and post-test scores (along with the difference between the two which we will refer to as the “improvement”).

The results show pre-test scores in all groups were around 4 out of 15, approximately the level of chance performance (lower than what we expected given the 4 trivial problems). Subjects in the RND and DRV conditions improve by another 3

\textsuperscript{6}Given the need for training data, the DBN subjects were necessarily run after running all of the FSM subjects.
points through training. Both RND and DRV groups perform at our hypothetical best case pre-test performance on the post-test. The EXP group gains almost 5 points through training. A standard $t$-test shows a significant difference ($p < 0.01$) in post-test score between the EXP and the other two tutors. The EXP tutor is clearly outperforming the other two.

But why are subjects performing at chance on the pre-test when prior knowledge of simple arithmetic, which we expect all of our subjects to have (given our university student subject pool), should lead to higher scores? To answer this, we look again at test scores restricting our focus to the trivial problems (the first 4 problems from Table 4.4). Table 4.9 shows the surprising results.

In all conditions, subjects are answering approximately 3 out of 4 trivial problems correctly on the pre-test, less than what we would hope to see, but clearly not guessing. However, on the post-test, these scores are reduced by approximately half in the EXP condition and by two-thirds in the RND and DRV conditions! Students actually perform worse on the trivial problems after training even in the EXP case! The effect is perhaps a bit weaker in the EXP condition (no significant difference), but clearly something is going wrong in all cases. It appears that subjects lose confidence in their prior knowledge during the training process!

To explain this, we note that subjects are never given a chance to verify that their prior knowledge does in fact apply. That is, they answer trivial problems on the pre-test with no feedback on individual problems and they are never exposed to trivial problems during training. We speculate that, giving a small number of trivial problems during training (when they do receive feedback on each problem)

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Pre-test</th>
<th>Post-test</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>28</td>
<td>4.21 (0.29)</td>
<td>7.18 (0.47)</td>
<td>2.96 (0.57)</td>
</tr>
<tr>
<td>DRV</td>
<td>22</td>
<td>3.95 (0.33)</td>
<td>6.86 (0.58)</td>
<td>2.91 (0.57)</td>
</tr>
<tr>
<td>EXP</td>
<td>23</td>
<td>4.52 (0.37)</td>
<td>9.35 (0.63)</td>
<td>4.83 (0.64)</td>
</tr>
</tbody>
</table>

Table 4.8: Experiment I: Mean Test Scores (with standard error)
Table 4.9: Experiment I: Mean Test Scores on Trivial Problems (with standard error)

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Pre-test</th>
<th>Post-test</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>28</td>
<td>3.36 (0.16)</td>
<td>1.43 (0.26)</td>
<td>-1.93 (0.32)</td>
</tr>
<tr>
<td>DRV</td>
<td>22</td>
<td>3.09 (0.24)</td>
<td>1.14 (0.28)</td>
<td>-1.95 (0.32)</td>
</tr>
<tr>
<td>EXP</td>
<td>23</td>
<td>3.30 (0.21)</td>
<td>1.83 (0.29)</td>
<td>-1.48 (0.27)</td>
</tr>
</tbody>
</table>

will allow subjects to verify their prior beliefs, leading them to maintain their pre-test trivial problem scores on the post-test. In the next experiment we pursue this idea.

It is interesting that the pre-test trivial problem performance implies that subjects perform worse than chance on the non-trivial pre-test problems. Although they have no way of reasoning about the answers to the non-trivial problems, subjects are not strictly guessing, but are applying some kind of flawed logic.

In this first experiment we have established that RND training in the Laff domain leads to only minimal improvement in test scores and that a “reasonable” fixed training policy (EXP), leads to a larger improvement in test scores.

4.5.2 Experiment II

Experiment II compares the performance of two tutors: FSM (Section 4.2.4) and DBN (Section 4.3). Recall that these tutors differ from those in the previous experiment in that they may repeat problems. In this experiment, we also introduced a number of trivial problems to the problem set (see Section 4.4.2) in order give students an opportunity to validate the transfer of prior beliefs about arithmetic to the Laff domain.

The Trivial Issue

Returning to the curious performance drop on trivial problems, we look at Table 4.10 which shows scores on the trivial test questions for all conditions. Both FSM and
<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Pre-test</th>
<th>Post-test</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>28</td>
<td>3.36 (0.16)</td>
<td>1.43 (0.26)</td>
<td>-1.93 (0.32)</td>
</tr>
<tr>
<td>DRV</td>
<td>22</td>
<td>3.09 (0.24)</td>
<td>1.14 (0.28)</td>
<td>-1.95 (0.32)</td>
</tr>
<tr>
<td>EXP</td>
<td>23</td>
<td>3.30 (0.21)</td>
<td>1.83 (0.29)</td>
<td>-1.48 (0.27)</td>
</tr>
<tr>
<td>FSM</td>
<td>26</td>
<td>2.81 (0.29)</td>
<td>1.96 (0.29)</td>
<td>-0.85 (0.26)</td>
</tr>
<tr>
<td>DBN</td>
<td>26</td>
<td>3.35 (0.25)</td>
<td>2.62 (0.29)</td>
<td>-0.73 (0.33)</td>
</tr>
</tbody>
</table>

Table 4.10: Experiments I & II: Mean Test Scores on Trivial Problems (with standard error)

DBN follow the pre-test trend of the earlier tutors. Among the post-test scores on trivial problems the DBN condition scores highest. The improvement scores for FSM and DBN show a different pattern from the earlier experiment. Comparing the difference in improvement between the different groups we find a significant difference between DBN and RND ($p < 0.05$, Tukey’s HSD). We suspect that with larger sample sizes both DBN and FSM would show significant differences from the other three conditions. It appears that introducing trivial problems to training does mitigate the decrease in score on trivial problems and we will now let this issue rest and move on to the primary analysis.

**Test Score Improvement**

Table 4.11 shows test scores for the two new tutors alongside those from the prior experiment. At first glance, we notice that pre-test scores are on par with those from Experiment I. The post-test score of the FSM is better than RND, but not as good as EXP. However, notice that the improvement score for FSM is quite close to that of the EXP. The DBN post-test and improvement scores are both very close to those of EXP, with the DBN improvement score slightly higher. From this cursory look, it appears that the DBN has matched, but not exceeded, the performance of the handcrafted EXP tutor and is perhaps a bit better than the handcrafted FSM policy. A standard $t$-test shows a significant difference ($p < 0.05$) between the improvement
scores of the three better tutors (DBN, FSM, and EXP) and the two poor quality tutors (RND and DRV). Using Tukey’s HSD, only the DBN is significantly better than RND and DRV. We have no evidence that the DBN’s performance is worse than either of the 2 expert handcrafted policies.

One concern that may be raised is the fact that the DBN was trained using data gained from students trained by the EXP and FSM policies and therefore may just be mimicking those policies. However, an examination of the actual policy employed by the DBN shows an interesting difference. Unlike the EXP and FSM policies, the learned DBN policy broke the links within problem chains during presentation. The DBN taught most of the primary problems first (in order of the easiest to learn based on the data), then the secondary and the more difficult tertiary problems. This tendency reflects the DBN’s preference to teach easy skills first (an effect of the discounted value function), a trait that is crucial in domains with a finite amount of training time (here, number of presentations). The DBN’s policy was far more successful than the fixed DRV ordering which also tried to swap between chains.

**Resampling**

Since we were limited to relatively small sample sizes we further explore the data using resampling. By selecting with replacement from the FSM sample we generated a new sample of 26 subjects, FSM'. We performed the same operation on the DBN

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Pre-test</th>
<th>Post-test</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>RND</td>
<td>28</td>
<td>4.21 (0.29)</td>
<td>7.18 (0.47)</td>
<td>2.96 (0.57)</td>
</tr>
<tr>
<td>DRV</td>
<td>22</td>
<td>3.95 (0.33)</td>
<td>6.86 (0.58)</td>
<td>2.91 (0.57)</td>
</tr>
<tr>
<td>EXP</td>
<td>23</td>
<td>4.52 (0.37)</td>
<td>9.35 (0.63)</td>
<td>4.83 (0.64)</td>
</tr>
<tr>
<td>FSM</td>
<td>26</td>
<td>3.70 (0.38)</td>
<td>8.38 (0.56)</td>
<td>4.69 (0.58)</td>
</tr>
<tr>
<td>DBN</td>
<td>26</td>
<td>4.08 (0.33)</td>
<td>9.35 (0.47)</td>
<td>5.27 (0.44)</td>
</tr>
</tbody>
</table>

Table 4.11: Experiments I & II: Mean Test Scores (with standard error)
group to give us $DBN'$. We then calculated the ratio of the improvement values for the two new samples as $\delta_{DBN'}/\delta_{FSM'}$ and repeated this process for 100,000 bootstrapped samples. If the means of the two groups are equal we would expect the ratio to be 1.0, but the upper and lower 2.5% quantiles on the bootstrapped samples yielded ratios of approximately 89% and 128% implying that the performance of the subjects under the DBN policy is at 89%-128% of the performance of those under the FSM policy. We applied the same procedure to compare the DBN to EXP, resulting in a range of 79%-115% which implies that the DBN is likely the same or slightly worse than EXP.

4.6 Discussion

4.6.1 General Overview

In this pilot study, we have given preliminary evidence that a skill teaching tutoring system can be designed using our DBN template (Chapter 3) and can be trained using data gathered from interactions with human subjects. We have shown that the performance of such a system, within a very restricted domain, is indistinguishable from the performance of algorithmic tutoring based on human expert opinion. Our next step will be to explore an expanded learning domain, removing many of the restrictions imposed in the pilot study.

4.6.2 Lessons Learned

Although we originally considered the finite field arithmetic domain to depend on a specific set of skills, we ultimately found it very difficult to distinguish what skills the students were actually learning. In the end, it seemed that students were mainly memorizing facts. We believe that “normal” subject matter studied by human students is usually reasonably separable into subtopics that have some sort of hierarchical dependency structure. That is, some topics have natural prerequisites; knowledge of some topics is clearly helpful (and often required) toward learning other more advanced topics. For example, it would be hard to imagine learning calculus
before having a basic understanding of algebra. The implication is that since natural subjects have prerequisites there are better and worse learning trajectories. We would like to explore a subject that has a clearly defined prerequisite structure in which students can apply knowledge they learn in earlier stages to help them in learning more advanced material.

A second issue with this study was the influence of prior knowledge. The majority of the skills required by the task were already known by students (e.g. rules like $X + 0 = X$.) It would be preferable to use a completely unfamiliar task. We also notice a general distaste among students for the mathematical nature of the finite field arithmetic task; a more palatable subject should have a positive effect on engagement. We address these concerns in the BLAST study in the following chapter.
In the case study detailed in this chapter, we approach the problem of teaching the skills required to understand a new language. We describe here a set of experiments in which students acquired the skills necessary to understand phrases in an artificial language\(^1\).

In the first section we describe the artificial language and the structure of the tutoring environment. In Section 5.2 we describe two hand crafted tutors used for baseline performance comparisons, and in Section 5.3 we describe the tutor learned by our system. Section 5.4 gives the details of the experiments and in Section 5.5 we provide the results and analyses. Finally, in Section 5.6, we discuss the results and provide a number of suggestions for future experiments of a similar nature.

5.1 Domain Description: BLAST

5.1.1 An Artificial Language

In Section 4.6.2 of the previous chapter, we described a number of shortcomings of the LAFF domain we used in our initial set of experiments that we must address in order to capture the essence of skill learning in a simple research domain. Namely, we desire subject matter that contains a clearly identifiable skill hierarchy, we would like to reduce or completely remove all prior knowledge, and we want something students will find enjoyable or at least not uncomfortable to learn (i.e. no math). Given these demands, rather than search for a subject that matches our needs, we applied this knowledge in the design of a new domain in which students learn how short phrases are constructed in an artificial language. As explained in Section 1.4.5 of Chapter 1,

\(^1\)Much of the material in this chapter is based on work published in Green et al. (2011a,b).
our main interest is the teaching and learning of skills in general and we treat the design of this domain in terms of skills rather than in any language specific way. To reduce individual variability and to simplify the experimental process, we build a domain with relatively few skills which can be learned within a one hour training session. We believed that a language domain would feel more natural to students and we found that in general students did indeed find learning in this domain to be a pleasant experience.

The use of a newly designed domain allowed us to avoid the confounds associated with students’ prior knowledge and expectations in familiar domains when evaluating the tutoring policies. Our interest is skill learning in general and this domain focused on learning the syntax and semantics rules in a small artificial language called Muq-Duq. The content of the language covers description of simple scenes consisting of geometric shapes of various colors. To concentrate on the learning of skills rather than on simple fact memorization, the size of the vocabulary in the language was severely restricted. The language comprises very few words, but these can be ordered to construct many phrases with very different meanings. The skills to be acquired are built into the phrase construction rules. The underlying intention in the construction of the skill set was to ensure that knowledge of some skills would “catalyze” the learning of other skills. That is, there is an inherent hierarchy in the skill set such that skills cannot be learned in isolation, making the design of teaching curricula necessary. Although quite small (designed to be learnable within a one hour period), the domain is sufficiently rich to distinguish performance of different tutoring policies. There is evidence in the literature that there are more “natural” orders of acquisition of grammatical structures (e.g. Michaud et al., 2001; Larsen-Freeman, 1976) giving us further evidence that an artificial language will be useful as a skill teaching testbed domain. We plan to implement this work on a larger scale in real world tutoring domains (not necessarily language related) after verifying that we are able to improve skill learning by planning.
5.1.2 Muq-Duq: The Language Defined

Students will learn a set of rules about how phrases are formed in a partial artificial language, affectionately referred to as “Muq-Duq”. Muq-Duq has only nine words, divided into three types of words: nouns, color-modifiers, and quantity-modifiers.

Nouns, (N)

Each of three nouns (N) refers to a simple geometric shape as follows:

\[
\begin{align*}
    bap &= \square \\
    muq &= \triangle \\
    fid &= \bigcirc
\end{align*}
\]

Color Modifiers, (C)

Three color-modifiers (C) are used as suffixes to modify nouns (i.e. NC).

\[
\begin{align*}
    duq &= \text{orange } \bullet & (e.g. \text{ muq duq } &= \bigtriangleup) \\
    nef &= \text{green } \bullet & (e.g. \text{ muq nef } &= \bigtriangleup) \\
    rop &= \text{blue } \bullet & (e.g. \text{ muq rop } &= \bigtriangleup)
\end{align*}
\]

Quantity Modifiers, (Q)

Each of the three quantity-modifiers (Q) is polysemous, having the following general meanings:

\[
\begin{align*}
    oy &= \text{ small, one, light} \\
    op &= \text{ large, many, very} \\
    ez &= \text{ not, none, non}
\end{align*}
\]

The specific meaning of a quantity-modifier depends on the context. Quantity-modifiers can be used in three different positions:
1.) as a prefix to an N to signify the size of the N (i.e. QN)

\[ \text{op } \text{muq} = \blacktriangle \quad \text{i.e. “large triangle”} \]
\[ \text{oy } \text{muq} = \blacktriangle \quad \text{i.e. “small triangle”} \]
\[ \text{ez } \text{muq} = \blacksquare \quad \text{or} \quad \bullet \quad \ldots \quad \text{i.e. “non-triangle”} \]

2.) as a suffix to an N to signify the cardinality of the N (i.e. NQ)

\[ \text{muq } \text{oy} = \blacktriangle \quad \text{i.e. “one triangle”} \]
\[ \text{muq } \text{op} = \blacktriangle \quad \blacksquare \quad \text{i.e. “many triangles”} \]
\[ \text{muq } \text{ez} = \bullet \quad \blacksquare \quad \text{i.e. “no triangles”} \]

3.) as a suffix to a C to signify the intensity of the C (i.e. CQ)

\[ \text{muq } \text{duq } \text{op} = \blacktriangle \quad \text{i.e. “very orange triangle”} \]
\[ \text{muq } \text{duq } \text{oy} = \blacktriangle \quad \text{i.e. “lightly orange triangle”} \]
\[ \text{muq } \text{duq } \text{ez} = \blacktriangle \quad \text{or} \quad \blacktriangle \quad \ldots \quad \text{i.e. “non-orange triangle”} \]

Multiple quantity-modifiers can be used in a single phrase:

\[ \text{muq } \text{nef} = \blacktriangle \quad \text{i.e. “green triangle”} \]
\[ \text{op } \text{muq } \text{nef} = \blacktriangle \quad \text{i.e. “large green triangle”} \]
\[ \text{op } \text{muq } \text{nef } \text{oy} = \blacktriangle \quad \text{i.e. “large light-green triangle”} \]
\[ \text{op } \text{muq } \text{op } \text{nef } \text{oy} = \blacktriangle \quad \blacktriangle \quad \blacktriangle \quad \text{i.e. “many large light-green triangles”} \]

The use of the quantity-modifier for negation can create phrases which are quite difficult to parse:

\[ \text{ez } \text{muq } \text{op } \text{nef } \text{oy} = \blacksquare \quad \bullet \quad \bullet \quad \text{“many light-green non-triangles”} \]
\[ \text{op } \text{muq } \text{op } \text{nef } \text{ez} = \blacktriangle \quad \blacktriangle \quad \blacktriangle \quad \text{“many large non-green triangles”} \]
We consider the fourth of these phrase examples to be overly perplexing and accordingly we disallow “ez” as a suffix to N in our experiments.

Valid Phrases

A valid phrase must contain an N or a C and must follow the rules given in previous sections, giving the following list of valid phrase structures:

1.) 1-segment phrases:

   N
   C

2.) 2-segment phrases:

   NC
   QN
   NQ
   CQ

3.) 3-segment phrases:

   QNQ
   QNC
   NQC
   NCQ

4.) 4-segment phrases:

   2quantity-modifiers can not stand alone (e.g. we have no way to depict “largeness” in isolation).
5.) 5-segment phrases:

QNQCQ

5.1.3 Acquiring Proficiency in Muq-Duq

Proficiency in the language requires knowledge of a set of atomic facts, a set of syntactic rules about how to properly connect the atomic facts, and a set of semantic rules about how the syntactic structure of a phrase affects meaning:

1.) Atomic Facts:

N: the set of nouns
C: the set of colors
Q: the set of quantity words

2.) Syntactic Rules:

NC: color-modifiers are only added as a suffix to an N
QN: a quantity-modifier may prefix an N
NQ: a quantity-modifier may suffix an N
CQ: a quantity-modifier may suffix a C

3.) Semantic Rules:

NC: a color-modifier suffix to an N designates the color of the N
QN: a quantity-modifier prefix to an N signifies the size of that N
NQ: a quantity-modifier suffix to an N signifies the number of N’s
CQ: a quantity-modifier suffix to a C gives the saturation of the C
As discussed in Section 5.1.1, the main intention during the design of the language domain was to ensure that a pre-requisite hierarchy existed such that knowledge of certain language components (i.e. proficiency in certain skills) would greatly aid in learning other components. From the facts and rules presented here and the valid phrases defined in Section 5.1.2 we can derive such a dependency hierarchy. In the graph shown in Figure 5.1, the white nodes indicate atomic $F$, each of the grey nodes represents a valid phrase structure from Section 5.1.2, and each edge indicates an intended prerequisite dependency. The assumption is that having knowledge of the meaning and usage of phrases toward the top of the graph will help students when learning about the phrases that depend on them (i.e. are beneath them in the graph), and conversely, lack of knowledge of pre-requisite structures should hinder learning. We consider the grey nodes in Figure 5.1 to be the set of skills $S$ that subjects will learn during training. Each skill $s \in S$ has a set of dependencies $D(s)$ whose members are the immediate “pre-requisite” skills of $s$ indicated by arrows in the figure.

### 5.1.4 Teaching Actions

The BLAST domain defines a set of teaching actions $A$ appropriate for teaching the language skills $S$ described in Section 5.1.3 and $A(s)$ is the set of all actions $a \in A$ that can be used to teach $s$. As in the LAFF experiments, students will mainly learn Muq-Duq by solving problems; however, in BLAST we further divide $A$ by introducing hints as a type of teaching action. The actions available to the tutor are to either present a hint about a skill $H(s)$, or to ask the student to solve a problem about the skill $P(s)$. Together $H$ and $P$ make up the complete set of all possible teacher actions in the domain. Tutors are not able to choose the exact problem or hint to present, they simply select the skill to teach and whether it should be presented as a hint or a problem. The actual grounded instance of the action is generated randomly from all available actions (whether hint or problem) about $s$.

We now describe the two types of actions in detail.
Multiple Choice Problems (MCP)

Each multiple choice problem will be posed in one of three forms. Type M problems present a picture (or *meaning*) and a set of phrases as answer choices:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>▲ = ?</strong></td>
<td></td>
</tr>
<tr>
<td>a.)</td>
<td><em>muq op</em></td>
</tr>
<tr>
<td>b.)</td>
<td><em>op muq duq</em></td>
</tr>
<tr>
<td>c.)</td>
<td><em>oy muq nef</em></td>
</tr>
<tr>
<td>d.)</td>
<td><em>muq op duq</em></td>
</tr>
</tbody>
</table>

Type W problems are phrases (*words*) with pictures as answer options:
Type F, or “fill in the blank” problems equate a phrase and a picture leaving out a word; the answers are words that, when filled in, make the phrase correctly describe the picture:

<table>
<thead>
<tr>
<th>fid duq = ?</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.) ▲</td>
</tr>
<tr>
<td>b.) ■</td>
</tr>
<tr>
<td>c.) ▲</td>
</tr>
<tr>
<td>d.) ○</td>
</tr>
</tbody>
</table>

As seen in the examples here, all multiple choice problems always have four choices. In the current set of experiments, the tutor is not able to select the type of multiple choice problem to present, only the skill it will teach.

A student’s answer to a multiple choice problem is always followed by a feedback message displaying the question along with the correct answer and whether the student’s answer was correct or not.

Hints

In addition to problems, in this study we introduce hints in the form seen in the following example:

| fid duq is ○ |

5.1.5 The Tutoring System

We built an Intelligent Tutoring System (ITS) called BLAST to teach the language just described. Training in BLAST consists of a sequence of learning events, each
Table 5.1: Definitions: $N$ and $C$

initiated by the tutor. The tutor for a given student is selected before beginning training and the same tutor is used throughout that student’s session. In Sections 5.2 and 5.3 we describe the various tutors used in our experiments.

Each learning event begins with the tutor’s selection of a skill to teach paired with an event type (MCP, Hint). The system generates a random event of the type and skill selected. In the case of a multiple choice problem, the system randomly selects the form (M, W, or F) of the problem. The system then presents the Hint or MCP to the subject. In the case of an MCP, the subject is allowed to view the problem for as long as he likes before selecting an answer. After answering, the subject is presented with a feedback message which he is again allowed to view for as long as he likes before continuing to the next event. In the case of a hint, the subject is allowed to study the hint for as long as he likes before moving on to the next event.

Since we are interested in skill learning (and not simple rote memorization), we do not wish for our subjects to spend time (which in our case is limited to 1 hour) memorizing atomic facts. Accordingly, we supply the students with a table of definitions for “vocabulary” words. A table of definitions for all of the $N$ and $C$ words (as shown in Table 5.1) was given to all students and was visible at all times during training and testing.

5.1.6 Evaluation

In order to measure the the performance of the various tutors we designed the test shown in Table 5.2. The test consists of a sequence of 20 multiple choice problems intended to broadly test proficiency in Muq-Duq, as defined in Section 5.1.3. Test
questions are administered in the order given in the table, presented as in training but without the feedback message. Each row in the table shows the main skill (see Figure 5.1) being tested, the type of multiple choice problem (see Section 5.1.4), the MCP itself and its 4 answer options with the correct answer highlighted. The same test is used regardless of experimental condition and there is no time limit imposed during testing. The number correct out of 20 is displayed to the student at the termination of each test and this is the only feedback received about a test. The test is intended to be repeated by each student at equally timed intervals during training to allow experimenters to track and analyze learning progress.

5.2 Handcrafted Tutors

A set of handcrafted tutors were designed for comparison against learned tutoring policies and for data collection to train learned policies.

5.2.1 Random Tutor (RND)

We established baseline student performance by using a tutor which selected learning events at random. The tutor randomly selected a skill to teach and also randomly decided whether the event was a hint or a MCP. That is, this tutor was blind to the student’s actions and performance. If there are no dependencies among training problems then training order should be irrelevant and a random tutor should perform as well as any other tutor. If there are strong dependencies we expect to see very poor performance under the random tutor and we might expect to see essentially random performance (i.e. guessing.)

State Space

Since the random tutor ignored the student there was no need to represent student state in this case.
<table>
<thead>
<tr>
<th>#</th>
<th>Skill</th>
<th>Type</th>
<th>Question</th>
<th>Option 0</th>
<th>Option 1</th>
<th>Option 2</th>
<th>Option 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>QN</td>
<td>MW</td>
<td></td>
<td>.</td>
<td>ez fid</td>
<td></td>
<td>oy fid</td>
</tr>
<tr>
<td>1</td>
<td>QN</td>
<td>MW</td>
<td></td>
<td></td>
<td>ez muq</td>
<td>ez bap</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NQ</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
<td>bap op</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>NQ</td>
<td>FB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>QN</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>QNQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>NQC</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>QNC</td>
<td>MW</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>NCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>NCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>QNQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>QNQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>QNQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>QNCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>QNCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>QNCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>NQCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>QNQCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>QNQCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>QNQCQ</td>
<td>WM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Definitions: $N$ and $C$
5.2.2 Expert Tutor (EXP)

We designed an algorithmic *Expert* tutor that made tutoring decisions according to a policy that a human tutor might be expected to follow. In designing the expert tutor we took into account the built-in skill hierarchy of the language. The EXP tutor taught the skills in an order based on the skill difficulty levels and dependencies as described in Figure 5.1. Easier skills were taught first followed by progressively harder skills. Within a group of skills of the same difficulty level the teaching order was somewhat arbitrary. However, since students are given the definitions in Table 5.1, we assumed that learning $NC$ would be easier than the other 2-word skills. Accordingly, we placed $NC$ as the first of the 2 word skills in the expert skill teaching order. The expert skill order$^3$ can be seen in Table 5.3.

**State Space**

The state space was defined by student proficiency$^4$ at the various skills (q.v. Figure 5.1). Proficiency on skill was represented by three discrete levels \{0, 1, 2\}, with 0 indicating an “unmastered” skill, 1 signifying partial mastery, and 2 representing a “mastered” skill. Each student’s complete state of knowledge was maintained as a vector with an entry for the student’s proficiency at each of the skills. For each subject, state was initialized to all 0’s. The 15 skills were taken in the expert presentation order with $Q$ inserted immediately after $C$. The Skill column in Table 5.4 shows the state space skill ordering and the Mastery column shows an example of a student that has the state “22100000000000” indicating mastery of both $N$ and $C$, some knowledge of $Q$, and no knowledge of any other skill.

The EXP tutor used Algorithm 1 to teach students and to update their state representations. In short, the algorithm does not introduce the next skill until the student has shown mastery of all previous skills in the list. The EXP tutor used proficiency update rules (q.v. Chapter 3, Section 3.1.1) to specify how $\rho(s)$ (the

---

$^3$Since $Q$ words cannot be presented alone, $Q$ cannot be taught in isolation and must be taught as part of another skill. Hence, $Q$ is excluded from this list.

$^4$We use “proficiency” and “mastery” interchangeably.
<table>
<thead>
<tr>
<th></th>
<th>Skill</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.)</td>
<td>N</td>
</tr>
<tr>
<td>1.)</td>
<td>C</td>
</tr>
<tr>
<td>2.)</td>
<td>NC</td>
</tr>
<tr>
<td>3.)</td>
<td>QN</td>
</tr>
<tr>
<td>4.)</td>
<td>NQ</td>
</tr>
<tr>
<td>5.)</td>
<td>CQ</td>
</tr>
<tr>
<td>6.)</td>
<td>QNC</td>
</tr>
<tr>
<td>7.)</td>
<td>NQC</td>
</tr>
<tr>
<td>8.)</td>
<td>QNCQ</td>
</tr>
<tr>
<td>9.)</td>
<td>NCQ</td>
</tr>
<tr>
<td>10.)</td>
<td>QNQC</td>
</tr>
<tr>
<td>11.)</td>
<td>QNCQ</td>
</tr>
<tr>
<td>12.)</td>
<td>NQCQ</td>
</tr>
<tr>
<td>13.)</td>
<td>QNQCQ</td>
</tr>
</tbody>
</table>

Table 5.3: Expert Skill Presentation Order
<table>
<thead>
<tr>
<th></th>
<th>Skill</th>
<th>Mastery</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.)</td>
<td>N</td>
<td>2</td>
</tr>
<tr>
<td>1.)</td>
<td>C</td>
<td>2</td>
</tr>
<tr>
<td>2.)</td>
<td>Q</td>
<td>1</td>
</tr>
<tr>
<td>3.)</td>
<td>NC</td>
<td>0</td>
</tr>
<tr>
<td>4.)</td>
<td>QN</td>
<td>0</td>
</tr>
<tr>
<td>5.)</td>
<td>NQ</td>
<td>0</td>
</tr>
<tr>
<td>6.)</td>
<td>CQ</td>
<td>0</td>
</tr>
<tr>
<td>7.)</td>
<td>QNQ</td>
<td>0</td>
</tr>
<tr>
<td>8.)</td>
<td>QNC</td>
<td>0</td>
</tr>
<tr>
<td>9.)</td>
<td>NQC</td>
<td>0</td>
</tr>
<tr>
<td>10.)</td>
<td>NCQ</td>
<td>0</td>
</tr>
<tr>
<td>11.)</td>
<td>QNQC</td>
<td>0</td>
</tr>
<tr>
<td>12.)</td>
<td>QNCQ</td>
<td>0</td>
</tr>
<tr>
<td>13.)</td>
<td>NQCQ</td>
<td>0</td>
</tr>
<tr>
<td>14.)</td>
<td>QNQCQ</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.4: Expert State Space Example
proficiency/mastery level at a given skill \( s \) is updated during a subject’s training. Solving three problems of a given skill correctly with no intervening errors on problems of that skill resulted in an increase in \( \rho(s) \) up to a maximum value of 2, while one incorrect answer caused a skill mastery decrement with a minimum of 0. Notice also that the algorithm allows mastery levels of dependency skills to be decremented due to repeated poor performance on a higher skill. In a case where a student has mastered all pre-requisite skills of \( s \), but continually answers problems of skill \( s \) incorrectly, leaving \( \rho(s) = 0 \), it is unreasonable to continue to bombard the student with problems from \( s \). Rather, it is assumed that the student’s difficulties are due to an incorrect labeling of pre-requisite skills as mastered. Decrementing \( \rho(s) \) for each skill \( s' \in \mathcal{D}(s) \) causes the tutor to re-teach those pre-requisites.

With the EXP tutor we expect to see performance significantly higher than that of the random tutor and we expect to see performance that is quite good.

5.3 Learned Tutor: DBN

The learned tutor was based on the template given in Chapter 3.

5.3.1 Network Structure

The state of our DBN consists of one factor for each skill \( s \in \mathcal{S} \) corresponding to the student’s proficiency \( \rho(s) \) on that particular skill. A number of intermediate level nodes are added to the basic 2-layer DBN structure to represent the effects of specific action instances on state transitions and to capture a number of correlations and interactions between various parts of the model. These intermediate nodes will also decrease the number of parameters in the DBN. For a full description of these specialized nodes see Chapter 3.

Since our goal is to teach as rapidly as possible the skills required to understand phrases in Muq-Duq, we base our reward function on skill proficiency levels. The reward function is simply \( R(\rho_t) = \sum_{s \in \mathcal{S}} w(s) \rho_t(s) \), that is a weighted sum of the number of skills that have been mastered so far by the student. In our studies we
Algorithm 1: Expert Tutoring Policy

\[
\text{skills} \leftarrow \{N,C,NC,QN,NQ,CQ,QNQ,QNC,QNCQ,QNCQ,\ldots\}
\]

repeat
  \textbf{if } \exists v \text{ in skills: mastery}[v] < 2 \text{ then}
  \text{ } s \leftarrow v
  \textbf{else}
  \text{ } s \leftarrow \text{random(skills)}
  \textbf{end if}

\textbf{if} answered incorrectly last 3 MCP from skill \textbf{s} \text{ then}
  \text{present HINT}
  \textbf{if} mastery[s] < 1 \text{ then}
    \text{increment mastery}[s]
  \textbf{end if}
\textbf{else}
  \text{present MCP and record answer}
  \textbf{if} 3 \text{ consecutive correct answers from skill } s \text{ and mastery}[s] < 2 \text{ then}
    \text{increment mastery}[s]
  \textbf{else if} 1 \text{ consecutive incorrect answers from skill } s \text{ then}
    \textbf{if} mastery[s] > 1 \text{ then}
      \text{decrement mastery}[s]
    \textbf{else}
      \text{decrement mastery of each immediate subskill of } s
    \textbf{end if}
  \textbf{end if}
\textbf{end if}
\textbf{end if}
\textbf{until} all \text{skills} are mastered, or time is up
always used \( w(s) = 1 \).

**Backbone Structure**

The basic structure of the DBN is as defined in Section 3.2 of Chapter 3. Figure 5.2 shows that structure for the BLAST domain. The grey circles at the top represent the skill proficiencies at time \( t \) and the grey circles at the bottom represent the same proficiencies at time \( t + 1 \). Notice that the \( N, C, \) and \( NC \) skills are not included in the graph. As explained before, we hard-coded the tutoring rules for those skills and once a student has shown mastery of those skills, they are always assumed to be mastered (i.e. they are never decremented.) The DBN policy is only used after mastery of the initial three skills. As shown in the figure, an individual’s proficiency at a particular skill has a direct influence on the same proficiency at the following time step.

The “Action” node in the figure represents the specific teaching action (q.v. Section 5.1.4) generated by the system (from the skill and event type selected by the tutor) for presentation at time \( t \).
Match Nodes

In a typical DBN the action node has links to each of the factors; however, in our case the action does not necessarily involve all of the factors. In fact, it usually involves only a small subset of them. Having a link from the action node to each factor introduces an increase in the number of parent values in the CPT for each factor, increasing the complexity of the search without adding any benefit.

When a tutor selects a skill and a learning event type (hint or multiple choice problem) the system itself randomly generates a grounded instance of that learning event. For every skill there are a large number of different possible multiple choice problems and hints. The system randomly fills in each “part of speech” slot in the selected skill with an actual Muq-Duq word creating a phrase to use as the solution to a multiple choice problem or the as the basis for a hint. The particular form of the phrase will dictate which nodes of the DBN are involved in the learning. If the particular action is about the skill $s$ of a given node (i.e. either a hint $H(s)$ or a problem $P(s)$) that node is likely involved in learning at that time step otherwise the action likely has no effect on the node. Including all links increases the size of each CPT by the total number of actions.

In Figure 5.3 each node at time $t+1$ connects to a middle layer node labelled “match”. These match nodes (large light blue nodes\(^5\) in the figure) are activated only when their corresponding skills are relevant to the current action. In this way the CPT’s are only increased in size by a factor of 3 rather than $2|S|$. For example, if the current action is a phrase of the form $QNQ$ then the match nodes for skills $QN$, and $NQ$ will be activated as well as the match node for $QNQ$.

As an example, the match node for $QNQ$ will be active when the particular action chosen by the tutor involves the presentation of a multiple choice problem or a hint related to phrases of the form $QNQ$. Figure 5.4 summarizes the behavior of the $QNQ$-match node by giving an example of what a CPT might look like for the $QNQ$ node. The first row highlighted in red shows the case where the action does

\(^5\) “op fid op rop oy”
Figure 5.3: DBN Action Matches

not match the skill. In this case, all other factors can be disregarded (indicated in the table as an “*”), $\rho(QNQ)$ is left unchanged (“nc” in the figure indicates “no change”). Similarly, the second row shows that in a case where the facts are not known there will be no change to the mastery level of $QNQ$. The following three rows show the effect of a hint matching $QNQ$. We use a deterministic rule, simply setting the mastery level to 1 if it is 0 or leaving it unchanged in any other case. The remaining rows show a few examples of cases where the action is a problem, matches the skill, and the facts are known. The probabilities are fabricated here, but reflect the idea that mastery of $QNQ$ will likely be retained but gaining mastery is a less likely event and depends strongly on knowledge of dependencies.

**Aggregate Nodes**

The fact that the skills required for mastery of the domains we are interested in have a pre-requisite hierarchy allows us to make another reduction in the size of the CPTs. By recognizing that it is difficult if not impossible to learn skills without having already mastered the majority of their pre-requisite skills we can aggregate a number of states. The *Aggregation* node counts the number of parent nodes with some value. For instance, the green Σ-Dep nodes in Figure 5.5 count the number of parent skills for a $\rho(s)$ that have not reached the mastery level. This allows nodes
that may depend only on the number of mastered pre-requisite skills to consider just this count, rather than all possible combinations, leading to an exponential decrease in the size of the corresponding CPTs.

**Fact Nodes**

While our goal in this work is to build policies for student skill acquisition, the ground problems we present also have facts in them. Some facts are assumed to be known a priori (since they are given to students as shown in Table 5.1) but a few of them are expected to be learned during the teaching session. We do not want to lower the proficiency rating on a skill if the facts used in the ground action were not known, since an unknown fact is likely to blame. Thus we introduce FactsKnown nodes for each type of action (shown as red nodes in Figure 5.6), which, when a ground action is presented, take on a binary value based on whether all the facts in the problem are known or not. When keeping track of fact proficiencies is prohibitive, or during the planning phase when answers to specific questions will not be tracked, one can estimate the probability of knowing a fact based on the number of skills known, since student progress and the number of learned facts are correlated, a technique we utilized in the planning component.
Figure 5.5: Aggregate Nodes

Figure 5.6 shows the connection between knowledge of atomic $Q$ facts and skills requiring that knowledge. That is, the middle layer nodes added in Figure 5.6 verify whether the $Q$ value(s) required by a particular skill is known. For example, the fact node connected to the $QNQ$ node matches proficiency at the atomic $Q$ facts at time $t$ to the particular $Q$ facts found in the current action. The $Q$ slots in the current $QNQ$ action could each be filled with any of the 3 $Q$-modifiers$^6$. The change in proficiency of a skill containing a $Q$-modifier depends not only on whether the current action makes use of the $Q$ slot relevant to that skill, but also whether the student has knowledge of the particular value presented in that $Q$ slot at time $t$.

Cascade Links

Another modeling challenge is that if the student answers a problem, multiple skill proficiencies might need to be updated, so we must ensure that these changes will be synchronized. These correlations are enforced by having all the skills that might be affected by such a change linked directly (in the same layer) to the next proficiency factor for the skill $s$. Such correlations can be seen in the bottom layer of Figure 5.7.

$^6$Section 5.1.2
Figure 5.6: Fact Nodes

Figure 5.7: Cascade Links
Figure 5.8: Full DBN

The Full Structure

Figure 5.8 shows the full DBN structure used, incorporating all of the features we have just described.

5.3.2 State Space

The state space for the DBN was defined in the same way as for EXP, by student proficiency at the various skills (q.v. Figure 5.1). Skill proficiency was again represented as three discrete levels \{0, 1, 2\}. Each student’s state of knowledge was maintained as a vector with an entry for the student’s proficiency at each of the skills taking the skills in the same order as the expert.

The proficiency update rules used by the DBN were nearly identical to those of the EXP. Solving three problems of a given skill correctly with no intervening errors on problems of that skill resulted in an increase in $\rho(s)$ up to a maximum value of 2, while one incorrect answer caused a skill mastery decrement with a minimum of 0. The DBN made use of fact proficiencies for mastery updates; it did not decrement a skill if it could blame the error on non-mastery of a particular fact.

As in the LAPF study, the DBN for BLAST was trained (i.e. the values in the CPTs were calculated) using data collected from subjects trained under a variety
of conditions and the details are given in Section 5.4.3. The DBN’s policy was learned from training data with the following exception. As stated earlier we have intentionally kept our domain size small for a number of reasons (q.v. Section 5.1.1). One of our reasons is to avoid extended computation time due to the state space of a DBN being exponentially large in the number of factors. Approximate planners exist that are effective at finding solutions and we will rely on them when we scale up. However, for the purpose of experimentation it is useful to keep complexity low and use a planning algorithm that finds an optimal policy (e.g. value iteration). For the sake of reducing complexity we hard coded the tutoring rule for the first 3 skills \((N,C,NC)\) and after a student showed mastery of these skills we never returned to them. The rule was simply to continue presenting a skill until mastery was reached and then move on to the next in the list. Since students were given a table of definitions covering the first two of these skills (see Table 5.1) mastery of the three was typically quickly achieved.

5.4 Experiment Details

5.4.1 Protocol

In this section we give an overview of the protocol used in the experiments, excluding details that are not relevant to the narrative. Additional protocol details can be found in Appendix B, along with a screenshot walkthrough.

To verify the effectiveness of the BLAST tutoring system we used the tutoring policies of each of the different tutors detailed in Sections 5.2 and 5.3 to train students and tested student understanding of the language at intervals during training. A number of different “meta” conditions were also applied to see how the various tutors compared under different tutoring constraints. The conditions are detailed in Section 5.4.2.

In all conditions, each student’s BLAST session consisted of four 7-minute training blocks with each block followed by the 20 question test presented in full in Table 5.2. The same test was always used (i.e. across all students and within each individual
student’s session) and there was no time limit imposed during testing. The number correct out of 20 was given to the student at the termination of each test and this was the only feedback received about each test. Students typically spent approximately 4 minutes on the 20 question test.

Within each 7-minute training block, the tutor’s task was to select the sequence of learning events to be presented to the student. Tutors selected the skill $s$ that the event should teach (see Figure 5.1) and the type, either hint $H(s)$ or multiple choice problem $P(s)$, of learning event to be presented (see Section 5.1.4). A specific grounded learning event from the chosen skill and type was then randomly generated and presented to the student. Hints were presented as the example shown in Section 5.1.4. Students were allowed to contemplate a hint as long as they cared to before moving on to the next event. Multiple choice problems were presented as described in Section 5.1.4. A response to the problem was requested from the student, but no time limit was imposed. After responding to a multiple choice problem, a simple feedback message was presented pointing out whether the student answered correctly or not and displaying the problem along with the correct answer. The subject was allowed to study the feedback, again without time limit.

The tutor was always aware of the skill and type of each problem since it chose them. It was also given access to the Muq-Duq words in the instantiated problem and a record of the number of consecutive times the student had answered problems of the given skill correctly or incorrectly. For example, in the case of an event covering the $NC$ skill the tutor would also know that the solution phrase to the problem instance was “$muq rop$” and that the student had answered the last $n$ MCP’s of type $NC$ correctly (or incorrectly). The various tutors made use of this information in their own ways.

As in the LAFF study, subjects for BLAST came from the University of Arizona Psychology Department subject pool. Students in the pool are allowed to select experiments to participate in from a list of currently open experiments on campus. Their decisions are based on brief descriptions of each study (usually a short paragraph) available on the department’s subject pool web site (see Appendix B for this
description).

From our previous experience with undergraduate students as human subjects we felt it was important to give the students some extra incentive to focus on the task. All students in the pool were awarded a modest amount of course credit for participating in the experiment; however, this credit was in no way dependent upon their attention to the given task. We wanted to reward good performance in some way to encourage students to apply themselves; however, since students were already receiving credit, department rules disallowed any additional tangible incentives. We were initially blocked by this restriction, but we feel that our final solution was quite elegant. We offered students what they most desired, which in the case of first year undergraduates was often simply to leave the experiment! As an incentive to apply themselves to the task, students were told “the system will let you leave early if you do well”. All students took the first and second tests. Any student who scored 18 out of 20 or higher on the second or any subsequent test was allowed to leave the experiment at that time.

Since we are interested in skill learning (and not simple rote learning) and our subject sessions are limited to one hour, we supplied the students with a table of definitions for “vocabulary” words. Definitions of all of the $N$ and $C$ words (as shown in Table 5.1) was given to all students and was visible at all times during training and testing.

5.4.2 Experimental Conditions

Using the RND, EXP, and DBN tutors, we compared student performance under a number of different conditions described in the following sections.

**Policy: Random, Expert, DBN**

Each of the tutors described in Section 5.2 and Section 5.3 provided us with a tutoring policy. We will refer to these policies as Expert (EXP), Random (RND), and DBN.
Choice vs No Choice

One of our research questions concerns the pedagogical value of choice in learning. Toward this end we added a modification to BLAST in which presentation of a learning event was always preceded by giving the student a choice between three possible options for the next event. This was done both to better engage the students and, when the choices were over different skills, to act as a form of human-guided exploration in the training. In the Choice version, the policy (EXP, RND, or DBN) was used to select a $P(s)$ or $H(s)$ as before. Rather than seeing a single event from that skill and type, in the Choice condition the student was shown three “topics” to select from. Each topic was a limited preview of the next event, showing only a picture (for M type problems), a phrase with or without a blank (for F and W type problems) or the words A HINT!. That is, the topics did not reveal the meaning in the hint or the answers to the multiple choice problems. The three topics were chosen at random by the system from the skill chosen by the tutor (e.g. if the tutor selected $NC$, the system presented three problems from the $NC$ category.) In cases where the event type chosen by the tutor was Hint, the system presented one $H(s)$ and two $P(s)$. The student selected one of these options and proceeded as before. We will distinguish the two versions of BLAST by referring to them as the Choice condition and the NoChoice condition.

Zone of Proximal Development

To further explore the value of choice in learning we introduced a variant of Expert Choice based on Vygotsky’s notion of the zone of proximal development (Vygotsky, 1978). In this Expert Choice ZPD condition, Choice was augmented by increasing the difficulty level of one of the 3 options presented to the student at each step. The difficulty was defined by the tutor’s (this condition was only run on EXP) estimate of the student’s current state of knowledge and our predefined difficulties (Figure 5.1). That is, to implement this we simply replaced one of the three options with a problem from the “next” skill to be taught (i.e. next in the Expert’s skill
ordering). This allows the student to have some choice about whether it is time to push forward. The intention was to allow students the opportunity to explore the horizon of their knowledge.

**RSA**

During analysis of some of the prior conditions it appeared that the “pace” of instruction might be influencing student performance and in order to examine this further we added a final condition in which we allowed the subject to take a stronger control of his learning trajectory. This condition was based on the Expert Choice condition, with the three preview options presented before each event. The middle option was always generated exactly as in the Expert Choice condition, i.e. from the “current” skill being taught. In this case however, the lefthand option was generated from the previously taught skill and the righthand option was from the next skill to be taught. Referring to the left option as *Review*, the middle as *Same*, and the right option as *Ahead* we call this condition Expert Choice RSA.

**Summary**

In summary, the experiments covered the following eight conditions:

<table>
<thead>
<tr>
<th></th>
<th>Policy</th>
<th>Choice</th>
<th>Variant</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.)</td>
<td>RND</td>
<td>NoChoice</td>
<td></td>
</tr>
<tr>
<td>1.)</td>
<td>EXP</td>
<td>NoChoice</td>
<td></td>
</tr>
<tr>
<td>2.)</td>
<td>DBN</td>
<td>NoChoice</td>
<td></td>
</tr>
<tr>
<td>3.)</td>
<td>RND</td>
<td>Choice</td>
<td></td>
</tr>
<tr>
<td>4.)</td>
<td>EXP</td>
<td>Choice</td>
<td></td>
</tr>
<tr>
<td>5.)</td>
<td>DBN</td>
<td>Choice</td>
<td></td>
</tr>
<tr>
<td>6.)</td>
<td>EXP</td>
<td>Choice</td>
<td>ZPD</td>
</tr>
<tr>
<td>7.)</td>
<td>EXP</td>
<td>Choice</td>
<td>RSA</td>
</tr>
</tbody>
</table>
The first three compare a baseline RND tutor, an EXP hand crafted tutor, and our learned DBN tutoring policy. The following three examine the effect of Choice on the three tutoring policies. The last two conditions take a more specific look at the effects of different types of Choice on the EXP tutoring policy.

5.4.3 Training/Learning the DBN Model

As we mentioned in Section 3.1.2 of Chapter 3, it is important to use a mix of training data collected under differing policies. The CPT parameters for BLAST were learned from student trajectory data collected during training in the following five conditions: RND Choice, RND NoChoice, EXP Choice, EXP NoChoice, and EXP Choice ZPD. The EXP Choice RSA data was collected after the DBN testing had finished and therefore was not used to train the DBN.

Training the CPTs requires a record of each subject’s transitions between proficiency states defined in terms of the skills represented by the DBN. In the LAFF study none of the tutors recorded this information explicitly and we had to convert all trajectories into this form before training the DBN. Here we perform the same operation on data collected under conditions using the RND policy. The EXP policy in BLAST already tracks student state transitions; however, the proficiency update rules which function well for human training are not the best rules to use for CPT training. The EXP policy incremented \( \rho(s) \) after 3 correctly answered problems of skill \( s \). If we train the DBN using these state transitions directly, we will be effectively ignoring all of the data in a trajectory that occurs on events where \( \rho \) is not changed. Training data is expensive and we would like to capitalize on any information it contains. In order for the DBN to make use of the more granular changes occurring on a problem by problem level, we modified the proficiency adjustment rules. All data collected in all five conditions was re-encoded using a proficiency adjustment rule that incremented \( \rho(s) \) after a single correct answer and decremented \( \rho(s) \) after a single error.
In this section we present a number of different analyses on different aspects of the experiments. Before proceeding we need clarify a relatively minor point. Recall from Section 5.4.1 that we allowed subjects to leave early if they received a score of 18/20 or higher on any test after the first. A small number of students did indeed “test out”. Table 5.5 shows the number of subjects in each condition and the number of those that tested out early. In some cases, this complicates analysis since these few subjects did not have the same number of training blocks and tests as the rest. In most of our analyses we use the common approach of last observation carried forward. For each analysis where this difference is relevant, we will clarify what was done regarding these few cases.

### 5.5.1 Evaluating the Domain: Can a Tutor Help You Learn Muq-Duq?

Figure 5.9 plots the mean proportion correct on each of the four tests for subjects in each of the eight conditions. Since the test is composed of multiple choice problems

---

7In the case of the few subjects that tested out their last test score was propagated forward as an estimate of their scores on skipped tests.
each with 4 choices, our expectation is a score of 0.25 if no learning has occurred and subjects are merely guessing at each answer. From the figure we see that in the RND Choice and RND NoChoice conditions, performance hovers around chance across all 4 tests. Under all of the other tutors, scores were close to chance on the first test but show a clear upward progression in subsequent tests. Under the non-random tutors, subjects are able to answer almost 45% of test questions correctly on the last test. Subjects trained by non-random policies improve significantly more from test to test than those in the RND conditions ($p < .0001$ in a multivariate analysis of variance).

These results establish the fact that learning in this domain does benefit from tutoring. In other words, the skills learned in this domain do have dependencies and the ordering of learning events is important. We do have a learning hierarchy; the domain is a sound test bed for general skill tutoring.
5.5.2 Evaluating the Learned Policy: Does the DBN Succeed?

We will next attempt to establish whether the DBN tutor achieves our goal of performing as well as or better than the expertly crafted tutor. For the comparisons in this section we will mainly restrict our view to the EXP No Choice and DBN No Choice cases as the pristine form of each policy, unsullied by human subject choices.

**DBN vs EXP: Test Performance**

Figure 5.10 plots the mean test scores for EXP No Choice and DBN No Choice with 95% confidence interval bars, and Table 5.6 reports the actual values plotted in the figure, along with the improvement in score from first to last test. Test performance under the two policies looks remarkably similar. The means do not differ significantly between the two conditions on any of the 4 tests, nor do the mean improvement scores differ significantly.

The amount and rate at which students learn in the EXP and DBN do not seem very different. Students start at roughly chance performance on the first test
and are able to answer roughly 45% of the questions correctly on the final test. A two-way analysis of variance comparing policy (DBN and EXP) crossed with test number shows a main effect of test number ($p < 0.0001$) and no effect of policy nor any interaction effect. That is, students improve from one test to the next but whether they work under the DBN policy or EXP policy makes no difference to their improvement. We have found no evidence suggesting that DBN and EXP are different.

But can we show evidence that they are not different? A 95% confidence interval for the difference between the mean improvements of DBN No Choice and EXP No Choice is $[-0.12, 0.10]$. Pooling all conditions and again comparing DBN against EXP the 95% confidence interval for the difference between the mean improvements narrows to $[-0.06, 0.07]$. It seems that the difference between the two means is likely to be quite close to zero.

**Qualitative Differences Between DBN and other policies**

We have shown that subjects under the DBN policy perform very similarly to those trained under EXP, but as discussed in the **LaFF** experiment, we must verify that the DBN policy is not simply mimicking the policies that it learns from. To do this we compare the DBN policy to the EXP policy detailed in Section 5.2.2. The learned DBN policy differed from the handcrafted EXP policy in a number of ways. First, the DBN policy exploits the deterministic effect of hints by giving a hint whenever a $\rho(s)$ is at 0, thereby avoiding the loss of proficiency of the pre-requisite skills. More interestingly, the DBN policy chose a different ordering for teaching the skills. The EXP policy teaches all skills of a given phrase length $n$ before moving on to skills

<table>
<thead>
<tr>
<th>Condition</th>
<th>N</th>
<th>Test 0</th>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP NO Choice</td>
<td>40</td>
<td>0.304</td>
<td>0.316</td>
<td>0.393</td>
<td>0.443</td>
<td>0.139</td>
</tr>
<tr>
<td>DBN NO Choice</td>
<td>40</td>
<td>0.288</td>
<td>0.350</td>
<td>0.388</td>
<td>0.439</td>
<td>0.151</td>
</tr>
</tbody>
</table>

Table 5.6: Mean Test Score (proportion correct)
of phrase length $n + 1$. For example, under the EXP policy, subjects must show mastery of $NC$, $QN$, $NQ$ and $CQ$ before the policy will begin teaching $QNQ$. The DBN policy specifically groups length 2 and 3 skills where only a $C$ or $N$ is added to the phrase structure. For example, the DBN follows the teaching of $QN$ with $QNC$. Intuitively, this does not seem like a bad idea, although it may not be the most obvious approach to a human teacher.

**DBN vs EXP: Skill Mastery**

So far our hopes for an outstanding performance by the DBN have been somewhat dashed. Although we have seen performance from the DBN that is quite comparable to the EXP, we had hoped for much better. After a great deal of frustration, head scratching, and data analyzing we had the realization that we were overlooking an assumption we had made early on in the design process. The belief that high test scores should result from mastery of the domain’s skills seems inherently obvious. However, this relies very much on the definition of mastery being accurate, and on the test corresponding quite well with the skills being trained. Recall that we based our reward function on our definition of skill mastery, by summing up the mastery levels of each skill $s$ in $\rho$, which leads the DBN to focus its policy on achievement of full proficiency as quickly as possible. The DBN is focused on increasing student’s skill levels, not on increasing their test scores. Figure 5.11 plots the mean reward value (with 95% confidence intervals) of the proficiency state of subjects after their last learning event preceding each test (i.e. their skill proficiency state upon entering each test). It is immediately apparent from the figure that the DBN has been performing as requested. The DBN has learned a policy that is very effective at helping humans to increase their skill proficiency as defined by the experimenter.

Comparing mean reward between DBN and EXP on the state preceding each test shows a significant difference between the two in all four cases (on the first test $p < 0.01$ and in the last 3 cases $p < 0.0001$, two-tailed $t$-tests). In fact, the DBN NO Choice condition is significantly different from all other conditions (including DBN Choice) on the final reward score ($p < 0.0001$, Tukey’s HSD). Figure 5.12 repeats
With respect to reward scores, the DBN No Choice clearly outclasses the other conditions. In Section 5.6.2 we discuss ways to modify our reward function in order to align the DBN’s goals with the human goal of high test scores.

Summary

Given the foregoing results, we have established that a problem-oriented ITS which repeatedly decides which skill to present to a student, can learn how to teach. In particular, we have shown that our general framework can be used to learn a tutoring policy that is competitive with policies crafted by experts in a domain. We also have shown that such a system does not simply mimic policies under which training data was collected. Rather, it constructs novel approaches which may not have been obvious to a human teacher, but still perform at an equal level with the human designed expert policy. Although the DBN’s policy does not currently outperform
the EXP, which we certainly would prefer, it has passed our original requirement and holds much promise for future improvements. At this point we will restrain ourselves from any further defense of the DBN policy, and we move on to explore our educational research questions and some surprising results encountered there.

5.5.3 The Effect of Choice on Learning

Since it has already been shown that there is no learning occurring in the RND cases we exclude both random conditions from all statistical analyses in this section.

Overview Of All Conditions

In Figure 5.9, we have already seen the overall test score trend. To look at how much learning is occurring, Table 5.7 shows the mean “improvement” in test scores (the difference between first and last\(^8\) test scores) along with standard error and standard

\(^8\) i.e. the last test taken by each subject, not necessarily the 4th test
deviation. Under all conditions we see an increase of approximately 15 percentage points from the first to last test (i.e. 3 points on the 20 point test). The largest improvement occurs in the EXP Choice ZPD condition and the smallest in EXP NoChoice. In fact, we notice a slightly higher score in all of the Choice conditions over the NoChoice conditions. In Table 5.8 we pool all Choice conditions together and compare with pooled NoChoice conditions, showing the mean improvement for these two groups. Although this table suggests that there may be some small effect of Choice on learning amounting to a difference in improvement of 0.028 (i.e. 2.8 percentage points, or 0.55 of a point on the 20 point test), it is not a significant difference and there is no interaction effect between choice and time across the 4 test measurements.

Let us look again at the improvement scores. Table 5.9 shows the mean improvement along with the mean of the first and last test scores for each of the 6 conditions, followed by the same for the conditions pooled. Here we notice that
<table>
<thead>
<tr>
<th>Condition</th>
<th>First Test</th>
<th>Last Test</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP No Choice</td>
<td>0.304</td>
<td>0.443</td>
<td>0.139</td>
</tr>
<tr>
<td>EXP Choice</td>
<td>0.248</td>
<td>0.411</td>
<td>0.164</td>
</tr>
<tr>
<td>EXP Choice ZPD</td>
<td>0.260</td>
<td>0.454</td>
<td>0.194</td>
</tr>
<tr>
<td>EXP Choice RSA</td>
<td>0.266</td>
<td>0.432</td>
<td>0.166</td>
</tr>
<tr>
<td>DBN No Choice</td>
<td>0.288</td>
<td>0.439</td>
<td>0.151</td>
</tr>
<tr>
<td>DBN Choice</td>
<td>0.262</td>
<td>0.426</td>
<td>0.164</td>
</tr>
<tr>
<td>Conditions Pooled:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO Choice</td>
<td>0.296</td>
<td>0.441</td>
<td>0.145</td>
</tr>
<tr>
<td>Choice</td>
<td>0.258</td>
<td>0.431</td>
<td>0.173</td>
</tr>
</tbody>
</table>

Table 5.9: Mean of First, Last, and Improvement

although the improvement scores are slightly higher in the Choice conditions, the First Test scores are higher in the NoChoice conditions, suggesting that subjects in the NoChoice conditions improve more during the first training block than others\(^9\). One explanation for this may be that subjects in the NoChoice conditions tend to proceed more quickly during training and may have simply seen more problems before the first test than subjects in Choice conditions. To examine this, we compare the number of learning events preceding Test 0 in Choice versus NoChoice conditions (again excluding all random conditions\(^10\)). Table 5.10 reports the mean number of learning events before Test 0 for each individual condition followed by the value for pooled Choice and NoChoice conditions. We note that NoChoice subjects have on average 5.3 more learning events before their first test than do Choice subjects, a difference which is significant ($p < 0.0001$). It is not yet clear whether this 5 event

\(^9\)and perhaps causing us to regret not giving a pre-test. However, given the scores of subjects under RND tutoring it seems safe to assume that untrained subjects would score approximately chance on the test.

\(^10\)from this point onward we will refrain from pointing out that we are always excluding the random conditions
Table 5.10: Mean Number of Learning Events Before Test 0

<table>
<thead>
<tr>
<th>Condition</th>
<th># Events Before Test 0</th>
<th>SE</th>
<th>SD</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXP NO Choice</td>
<td>32.650</td>
<td>1.033</td>
<td>6.534</td>
</tr>
<tr>
<td>EXP Choice</td>
<td>26.250</td>
<td>0.729</td>
<td>4.612</td>
</tr>
<tr>
<td>EXP Choice ZPD</td>
<td>26.400</td>
<td>0.840</td>
<td>5.310</td>
</tr>
<tr>
<td>EXP Choice RSA</td>
<td>25.320</td>
<td>1.205</td>
<td>6.026</td>
</tr>
<tr>
<td>DBN NO Choice</td>
<td>31.225</td>
<td>0.724</td>
<td>4.577</td>
</tr>
<tr>
<td>DBN Choice</td>
<td>28.308</td>
<td>0.659</td>
<td>4.118</td>
</tr>
<tr>
<td>Conditions Pooled:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO Choice</td>
<td>31.938</td>
<td>0.632</td>
<td>5.651</td>
</tr>
<tr>
<td>Choice</td>
<td>26.688</td>
<td>0.418</td>
<td>5.021</td>
</tr>
</tbody>
</table>

lead could produce the difference in Test0 scores between the two conditions (0.038, or 0.75 of a point on the 20 point test).

To investigate, we next look for a positive correlation between number of events before Test0 and score on Test0. If such a correlation exists then we may have an explanation. We find no correlation whatsoever between event count and test score in the Choice condition. In the NoChoice condition we find an extremely weak positive correlation ($p = 0.045$, $r^2 = 0.05$). An analysis of variance (MANOVA) with one factor being Choice/NoChoice and the second test score on each of the 4 tests, there is no effect of condition nor is there any interaction effect between condition and test. Unfortunately, we are unable to say more about the overall effect of choice (but see the next section on gender) and we must leave this mystery for now.

5.5.4 Gender Differences in BLAST

We now look at some curious gender differences in performance. Table 5.11 shows the mean test improvement in each condition by gender. We immediately notice an overall difference. Males show an improvement of almost 21 percentage points
from first to last test, nearly twice that of the female mean improvement. Next look at the values in the Male column. Males seem to be unaffected by the different conditions, they improve by approximately 0.2 in all cases. Females on the other hand show a very interesting pattern. Recall from the previous section that having Choice appeared to have a very slight positive but not significant effect. It appears that the effect exists within the female subjects, it was masked earlier since genders were pooled. Females improve nearly 2.5 times (significant at the $p < 0.05$ level, two tailed $t$-test) more in Choice than in No Choice conditions!

Table 5.12 shows the mean test scores and mean improvement by choice condition and gender. Here the Choice condition pools the data from EXP Choice, EXP Choice ZPD, EXP Choice RSA, and DBN Choice conditions, while the No
Choice condition pools the EXP No Choice and DBN No Choice. Overall, women improve significantly less than men \((p < 0.0020, \text{ two-tailed } t \text{ test})\), and women do better in the two Choice conditions than in the two NoChoice conditions \((p < 0.046, \text{ two-tailed } t \text{ test})\). A two-way analysis of variance with Choice and Gender as factors shows a strong main effect of Gender \((p < 0.0006)\), no effect of Choice, and a marginal interaction effect \((p < 0.10)\). The No Choice Female group differs significantly from both Male groups with \(p < 0.0089\) in the No Choice Male and \(p < 0.0123\) in Choice Male (Tukey’s HSD). When we look at the last test score, rather than improvement, we see a strong main effect of Gender \((p < 0.0005)\), no effect of Choice and no interaction. Here we notice that the No Choice Male group scores significantly higher than both Female groups (No Choice Female \(p < 0.0148\), Choice Female \(p < 0.357\)). The No Choice Male group scores highest overall and the No Choice Female group scores lowest.

Across conditions, choice does not produce significantly higher test scores than a lack of choice. However, when choice is lacking, women do not improve from one test to the next, while men improve whether or not they have choice. Said differently, choice (or the lack of it) is associated with whether women do as well as men in the BLAST domain. A repeated measures multivariate analysis of variance (MANOVA) tells us that the average participant improves \((p < 0.0001)\) and the amount they improve depends on gender \((p = 0.0029)\) but not the choice condition \((p = 0.413, \text{n.s.})\). However, there is a marginally significant interaction between time, gender and the choice condition \((p = 0.054)\) suggesting that the effect of choice on a participant’s improvement over time depends on gender.

The questions raised by these results are, first, why does a lack of choice seem to depress women’s scores, and, second, is there any positive effect of choice on scores? To answer these questions we must first gain a better understanding of the effects of choice on learning.

As noted in Section 5.4.2, participants in the Choice conditions see three problems and select one. The main difference between the problems is their type, which may be a picture (type M), a sentence (type W), or a picture-sentence equation.
with one word missing (type F), as defined in Section 5.1.4. As it happens, the three problem types are not represented equally on the test. In fact, there is only one problem of type F on the test, eight of type M and eleven of type W. If participants recognize this after the first or second iterations of the test, then they might choose problems of type M and W, as 19 of 20 test problems are of these types, and neglect problems of type F. This would be a rational strategy.

In fact, some students do choose problems of one type preferentially, but not rationally. There are four iterations of the test, each preceded by one training block of problems. During the first training block, participants have no idea what they will see on the test. During the second, they might not be expected to remember what types of problems were on the test. But during the third and fourth training blocks, if they have good memories, they might remember the types of problems that are on the test and choose these types during training.

We define the tendency toward one type of problem as the proportion of times that a participant chooses that kind of problem during training block 3 or 4. Every participant therefore can be viewed as a point in 3-space, defined by his or her tendency toward problems of types M, W, and F. Note that participants in the No Choice conditions also have tendencies toward problems, but they are controlled by the BLAST system, which selects among the three types at random.

We clustered all the participants by their tendencies toward problem types using $k$-means clustering with $k = 2$. The centroids of these clusters tell us that some participants tend to favor a single problem type while others work on all problem types with roughly equal frequencies. The first cluster, which we call specialized (S) includes 50 participants, of whom 34 are in the Choice condition. In this cluster, participants spend 48% of their time with problems of type M, 27% with problems of type W and 24% with problems of type F. In the second cluster, which we call broad (B), participants spend 29%, 36% and 34% of their time with problems of types M, W and F, respectively.

These proportions are broken down by condition, gender and cluster in Table 5.13. The first thing to notice is that participants in cluster B generally improve
<table>
<thead>
<tr>
<th>Condition</th>
<th>Gender</th>
<th>Cluster</th>
<th>N</th>
<th>Type M</th>
<th>Type W</th>
<th>Type F</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choice</td>
<td>Female</td>
<td>S</td>
<td>18</td>
<td>0.52</td>
<td>0.24</td>
<td>0.23</td>
<td>0.10</td>
</tr>
<tr>
<td>Choice</td>
<td>Female</td>
<td>B</td>
<td>20</td>
<td>0.30</td>
<td>0.37</td>
<td>0.32</td>
<td>0.19</td>
</tr>
<tr>
<td>Choice</td>
<td>Male</td>
<td>S</td>
<td>16</td>
<td>0.49</td>
<td>0.27</td>
<td>0.23</td>
<td>0.14</td>
</tr>
<tr>
<td>Choice</td>
<td>Male</td>
<td>B</td>
<td>25</td>
<td>0.27</td>
<td>0.39</td>
<td>0.34</td>
<td>0.25</td>
</tr>
<tr>
<td>NoChoice</td>
<td>Female</td>
<td>S</td>
<td>10</td>
<td>0.44</td>
<td>0.30</td>
<td>0.26</td>
<td>-0.03</td>
</tr>
<tr>
<td>NoChoice</td>
<td>Female</td>
<td>B</td>
<td>28</td>
<td>0.30</td>
<td>0.33</td>
<td>0.36</td>
<td>0.09</td>
</tr>
<tr>
<td>NoChoice</td>
<td>Male</td>
<td>S</td>
<td>6</td>
<td>0.43</td>
<td>0.29</td>
<td>0.27</td>
<td>0.30</td>
</tr>
<tr>
<td>NoChoice</td>
<td>Male</td>
<td>B</td>
<td>35</td>
<td>0.31</td>
<td>0.35</td>
<td>0.34</td>
<td>0.20</td>
</tr>
</tbody>
</table>

Table 5.13: Improvement from the first test to the last by Condition, Gender, and Cluster. The average values for tendency toward problems of types M, W and F are also shown.

more than those in condition S. This is true whether or not the participants have choice. The reason, we believe, is that participants in cluster S focus on problems of type M to the exclusion of the others. Unfortunately, only eight of twenty problems on the test are of type M, and eleven are of type W, so specialization to problems of type M results in lower improvement. (In general, NoChoice participants are in the B cluster – only 16 of 79 are in the S cluster – because the NoChoice problem selection algorithm tries to present equal numbers of problems of type M, W and F.)

Consider now women in the NoChoice condition. Those in cluster S, who choose one type of problem over others, do particularly badly (improving −0.03%). But interestingly, women in the B cluster, who study all kinds of problems with roughly equal frequency, improve by 9%. Compare this with women in the B cluster in the Choice condition, who improve 19%. Both groups of women “get a broad education,” but those in the Choice condition improve more. However, nothing happens in the choice condition besides choosing between problem types, and that effect is already accounted for by clustering the participants into the specialized and broad clusters.
We are left with the conclusion that choice has two effects: One is the deleterious effect of specializing to one kind of problem at the expense of the others. The other is a psychological benefit of having choice. One can estimate this benefit by comparing improvement across comparable Choice and NoChoice conditions in Table 5.13. For example, the difference between women in cluster S in the Choice and NoChoice conditions is the difference between 10% and −0.03%, or 13%. We would expect these differences to be larger in the S cluster than the B cluster because, in the B cluster, Choice participants look more like the average NoChoice participant; that is, they are “broadly educated” whether or not they have choice. In fact, in the S cluster the psychological benefit of choice is 13% (for women) and −16% (for men, though beware the small number of men in the S cluster in the NoChoice condition); while it is 10% (for women) and 5% (for men) in the B cluster.

5.6 Discussion

5.6.1 General Overview

The Blast study has shown that our DBN template can be used to design a skill teaching problem oriented ITS. Training the DBN using data gathered from interactions with human subjects allows the system to generate a novel policy that performs on par with expertly handcrafted policies in many respects and better than expert policies in some. We have also shown that, while conducting artificial intelligence research on the design of an ITS and collecting data for use in that regard, we can concurrently run educational research experiments. We find that the two are quite inextricably tied and that showing good general results regarding the performance of the ITS is not sufficient. Researchers must be aware of the possibility of differential performance by subject group (e.g. gender). It may be the case that an ITS that performs perfectly well for one type of student will be entirely inadequate for another. As researchers, we must be alert to such possibilities. As is always the case, the Blast studies have left us with more questions than answers and we now mention a few of the more important.
5.6.2 Improving Test Scores

As we described in Section 5.5.2, there is a disconnect between our current reward function which is used by the DBN to evaluate student performance and our test which is used by humans to evaluate student performance. The DBN is clearly performing well at maximizing the reward scores for subjects, we would like to find a way to align the reward function with higher test scores.

Currently, the tutors are very limited in their available behaviors. It stands to reason that limiting a tutor’s actions will limit its ability to teach. We would like to experiment with incorporating more granularity in tutor decision making. Given the earlier discussion on types of problems being presented, it may be beneficial to allow the tutor to choose problem types (M, W, F) as well as skills. We would also like to introduce more accurate hints. Currently hints are simply generated at random from the skill chosen by the tutor. When a subject answers a problem incorrectly, it would be more informative to generate a hint that shows why the subject’s answer was incorrect.

It may be the case that a simple change to the proficiency adjustment rules could lead to better learning. A set of experiments designed to find the best rules would be useful. We may find that it is necessary to use dynamic rules that adapt to individual subjects. It would be interesting to structure the DBN to allow it to reason about the proficiency rules.

Finally, given the gender differences noted in the results section, it would be interesting to introduce a gender factor to the DBN to see if it learns qualitatively different policies. It is likely that under the current restrictions different policies would not result. In fact, we did attempt to train separate DBNs by dividing our human subject data by gender. The resulting policies were essentially equivalent. We believe that the DBN would require more factors, and perhaps a much larger training set to enable it to capture gender differences.
5.6.3 Understand The Gender Difference

Some prior research suggests that choice is associated with higher learner motivation and persistence. Lepper et al. (1993) observed skilled human tutors working one-on-one with low-performing students, and noted that tutors often constructed choice opportunities for students, apparently with the objective of maintaining students engagement. For example, a tutor might ask a student if he or she would like to review or move on to new, more challenging material.

Other research suggests that including choice in instructional software tends to sustain learner motivation, even if the choices are not directly related to the nature of the instruction. Cordova and Lepper (1996) studied the impact of choice on elementary school students who worked with instructional software for learning arithmetic. The software involved a game format in which the child advanced along a number line when he or she provided correct answers. In one condition, the child could choose whether the agent played optimally (“the best it can”) or more like a human competitor (“pretty well but not great”). Students who were able to make choices about software features that were actually irrelevant (e.g., whether they were a space traveler or a pirate) learned significantly more (based on a pre-post-test comparison) and indicated that they felt ready for a more challenging version of the game in the future (suggesting higher self confidence).

Positive results have also been reported for intelligent tutoring systems that allow learners to make choices, including the types of explanations that they would like to view, when to review solutions for missed problems, and when to move from one type of activity to another (Beal et al., 2010, 2007c). Interestingly, in both cases the strongest benefits were observed in students who initially described themselves as low on academic motivation (and whose teachers agreed). The results are consistent with the work on how human tutors incorporate choice into instruction to motivate disengaged learners. However, these studies did not include a direct comparison of pedagogical strategies that allowed or did not allow learners to make choices during instruction.
No gender differences were reported in the Cordova and Lepper (1996) study. However, gender was not included as a factor in several analyses where a possible interaction of choice with gender might have been revealed. Work by Woolf et al. (2006) found gender differences in students’ utilization of hints in an ITS that allowed students to choose whether or not to view a hint, and to choose the type of hint they wished to view (animated or more text-based) when learning to solve challenging mathematics problems. More specifically, females were more likely to activate the hints, spent more time viewing the hints than males, and had a more positive reaction to the ITS than males.

Although limited, the results regarding the benefits of choice for females are suggestive, but we have not found another direct comparison of versions of an ITS that included or did not include options for learner choice as we explored in the BLAST study.

Research on gender development does indicate that there are more individuals in the population of males who have a preference for high challenge and risk taking (Beal, 1994). Other research on academic motivation has found that females tend to be more realistic in their assessment of their own abilities, and can at times underestimate how well they are doing relative to how well they are actually doing. In contrast, males tend to be more optimistic and even unrealistic about how well they are doing. To our knowledge, there are no studies in this area of research that examine different effects of specific pedagogical strategies on males and females. However, it would not be unreasonable to hypothesize that males might have a higher tolerance for a pedagogical strategy that might be described as “aggressive” in the sense of continually presenting challenging items without opportunities for choice and review. The present study allowed us to investigate this hypothesis.

On its own, this finding may be of moderate interest in that it is somewhat consistent with prior findings suggesting that choice is especially helpful to students with less positive views about their ability. Women and men in this study did not differ in their initial performance and, because the domain to be learned was completely novel, there was no reason for women to have lower views of their ability
in the task. However, prior work would suggest that it is at least possible that women reacted differently than men as they progressed through the training and that the overall experience, to include the ongoing uncertainty that is inevitable in trying to figure out a novel and complex domain, was more aversive for them. Choice may have ameliorated these reactions, at least in part, and allowed women to feel more in control over the experience and, in fact, led to improved performance. The more subtle and, in our view, most interesting thing about the results is that the psychological benefits of choice for women seem to be independent of what they actually did in the choice condition. More specifically, seeing a broad range of problems is helpful but the benefits to women go above and beyond this effect.

The present study provides a direct experimental test of the effects of choice on learning. One conclusion might be that choice tends to help women on average and does not appear to hurt men, so incorporating ways for the learner to make choices should be generally beneficial. Interestingly, the present findings and those of Cordova and Lepper (1996) suggest that it is the experience of choice rather than what the choices actually are that provide the benefits for learners.

The fault is not with the idea of deriving policies from data – women fared no better in the hand-constructed ExpertNoChoice condition – but it does suggest that we cannot expect to learn effective policies from data without putting a good deal of thought into what makes students different from one another, and how educational actions might interact with these differences.

5.6.4 Introducing A Human Tutor

In the conditions studied here, we have compared the ITS performance to hand-crafted expert teaching policies. It would be interesting to compare the ITS to live human teacher performance. However, this would pose a number of challenges. In the BLaST domain, we purposefully conceal the set of rules which define the Muq-Duq language. Subjects are forced instead to infer the underlying rules through solving problems. Clearly, it would be much easier to teach students the language by simply having them read Section 5.1.2, but this would defeat our purpose of
using BLAST as a skill learning domain. In order to use live human tutors we would have to restrict them similarly and then monitor their performance to ensure that they followed the restrictions. A more plausible approach would be to introduce a “Wizard of Oz” scenario where a hidden human tutor is restricted to the same toolset that is available to the ITS by placing him behind an interface and leading the subject to believe that the computer was in charge.
CHAPTER 6

CONCLUSIONS AND FUTURE WORK

The previous chapters describe a template which we use for building problem-oriented skill teaching intelligent tutoring systems based on a Dynamic Bayes network framework. To do so we introduced a number of specialized structures to the DBN including \textit{Match} nodes, \textit{Facts Known} nodes, \textit{Aggregation} nodes, and \textit{Cascade} links. These additions help to reduce the size of CPTs, aid in extracting as much information as possible from our data sets, and allow the DBN to capture correlations between variables that would otherwise be missed. We presented two case studies in which the template was adapted to very different teaching domains, documenting in each case the process of building the DBN using the specialized structures we provide. We describe the data gathering process and how to train the ITS using that data. In both case studies, the performance of the ITS was validated through human subject experiments. The performance of students trained under the policy learned by our ITS is compared with the performance of students trained using policies crafted by human experts. The results of these studies show that our template is a viable technique for designing ITSs that teach in skill based domains.

We have also shown that, not only can we run educational research experiments concurrent with our ITS research, but that it is quite important that we do so. Our interest in the effect of choice on learning led us to the interesting discovery that our ITS had greater success with men than it did with women. Since the goal of the ITS is to provide high quality personalized education to every individual, this is clearly not an issue we can afford to overlook. We hope that our work serves to encourage other ITS researchers to take advantage of their own systems to add to the body of literature on human learning.
6.1 Future Work

6.1.1 Bringing Mastery into Alignment with Test Scores

In Section 5.5.2 (Chapter 5), we made the discovery that the DBN was merrily outperforming all other conditions with respect to the defined reward function, while the experimenters were frantically scouring the data to discover why the DBN was not outperforming the rest with respect to the test scores. In our next round of experiments we intend to design a reward function that more closely reflects the human expectation that mastery of training problems leads to high test scores. The fact that our current reward function (based entirely on skill mastery) is not a great predictor of test scores may indicate a mismatch between what subjects are learning during training and what they are being tested on. Or, it may indicate that our definition of “mastery” is simply insufficient; subjects need more practice at different skills than we are currently requiring. We can leave the reward function alone and focus on refining our definition of mastery, or we can modify the reward function to bias it toward strong test scores. The immediate thought is to somehow introduce test scores into the reward function. Since test scores are only available after each test, we can not directly use test scores as reward values. We would need some way to pair each proficiency state \( \rho \) with a test score, but that implies predicting test scores from proficiency states, an activity at which we have not yet shown great skill. Given the difficulties we have faced in detecting relations in the data, it may be prudent to focus on refining mastery before we attempt to inject test data into our reward function. We may simply need to adjust the current set of proficiency update rules to bring “mastery” of training problems into alignment with mastery of test questions. It may be that we need finer granularity in our mastery levels. A comparison study where conditions differ in their update rules would be quite useful. On the other hand, it may be time to upgrade our simplified approach to state tracking and use of the more complex state tracking algorithms described in the literature (e.g. Corbett and Anderson, 1995; Baker et al., 2008).
6.1.2 Using Virtual Students To Augment Collected Data

In a university research setting it is often quite difficult to gather large amounts of human subject data. In our study we were restricted to a few hundred subjects per semester. Although our template design does much to wring the maximum amount of information out of the data we collect, we were still faced with the challenge of sparse training data. In our experiments, we incorporate random tutors and teaching conditions that used human guided exploration in order to allow the DBN to investigate areas of the search space that would never be probed under normal human tutelage. Even so, there are great holes in our data. One possibility is to guide the learning agent using the principle of “optimism in the face of uncertainty” in order to explore the space more broadly. However, this standard technique would require a large amount of data (for human subjects) and requires manipulating students into specific situations where they know or don’t certain skills, a difficult challenge in itself. A possibility that we have only briefly examined is to use virtual students to augment the data we collect. We might consider breaking the ITS training process into cycles. Each cycle would begin by collecting data from human subject experiments and training the ITS using that data. Next we would calibrate a virtual student (or students) against the human subject data, making the virtual student performance match the human performance as closely as possible under the same training conditions. We could then run large numbers of training trials using virtual students with a random tutor in order to more widely explore the search space. This data could be used to augment the previously gathered human data in training a DBN for the next round of human subjects. We would hope to see improvement in each new group of human subjects to verify that the ITS was indeed making good use of the new data.

6.1.3 Resolving the Gender Mystery

In the BLAST study we described a curious gender difference in test performance that showed up as interaction effects between gender, policy, and choice condition.
Although we have made progress, we have not yet isolated the cause of the difference. It would be very interesting indeed if we could train 2 ITSs, one for women and one for men, and show that each gender performs well under its own policy and poorly under the other. In order to proceed with this, it is likely that we will need to modify our state representation to capture more details. We have already mentioned increasing the granularity of action choices available to the tutors by allowing them the choice of problem type and by introducing more complex hints. It is also likely that we will need much larger data sets.

6.1.4 Deployment To A Large Scale ITS

Given the success of our template in deriving teaching policies for our 2 case studies, which are purposefully limited in size, we would next like to verify that it works on a larger scale. In the future, we plan to deploy our system in a large-scale online tutoring setting. The AnimalWatch ITS Beal et al. (2010), which has proven to be effective in teaching algebra readiness mathematics, including basic arithmetic, fractions, variables and expressions, statistics and probability, and simple geometry (www.animalwatch.org) is one such possibility. With roughly 3000 interactions a day, collecting data to train (and re-train) our models in this setting could be done much faster than in our current human-subject studies.

6.2 Parting Comments

As technology expands to fill gaps in the world’s varied educational needs, AI will continually take a stronger role. Intelligent Tutoring Systems provide a way to share high quality education. Intelligent Tutoring Systems are designed to learn from data provided by previous students in order to adapt and modify themselves, continually improving their ability to provide high quality individualized instruction to each of their students. This dissertation has described a template for building a skill teaching ITS, along with verification of its efficacy in teaching real information to real human learners. We hope that the information we have provided here will aid other
researchers in pursuing our common goal of providing high quality, individualized instruction to anyone anywhere.
A.1 Introduction

The purpose of this experiment was to analyze the efficacy of different policies used by a tutoring system to teach students a new subject. The experiment compared the performance of students learning to solve problems in an unfamiliar domain. The domain is finite field arithmetic on a field of size four\(^1\).

The experiment compared student performance at learning to solve problems when the problems are presented according to each of the following presentation policies:

1. randomly ordered problems
2. a fixed order specified by an expert in the domain
3. a fixed order learned by a machine learning algorithm
4. a policy in the form of a finite state machine (FSM) designed by an expert in the domain
5. a policy learned by a machine learning algorithm

The data collected from the random presentation condition (1) was used to train the model used to determine the problem presentation order for condition 3. Data collected in conditions 1, 2, 3, and 4 was used to train the model that learned the policy used in condition 5.

\(^1\)that is a “Galois Field” of size 4, \(GF(2^2)\)
A.2 Method

A total of 128 test subjects were drawn from the University of Arizona INDV101/INTRO PSYCH SUBJECT POOL. Subjects were divided into the following groups:

1. random ordered presentation (30 students)
2. expert ordered presentation (23 students)
3. learned fixed order (22 students)
4. expert FSM presentation (26 students)
5. algorithm ordered presentation (27 students)

Students in all groups were required to read and sign a consent form (Appendix C, Section C.2) and were given the following instructions:

“This is the consent form. Please read it and initial each page front and back. Sign and date the bottom of the second page. The last sheet is for you to take notes on while you work through the problems to help you remember what you have learned. You have 1 hour which is plenty of time, so don’t feel rushed. At the end of the hour you will be debriefed about the experiment. Once you have read and signed the consent form, sign in and follow the instructions you are given.”

“Arithmetic in finite fields is in some ways similar to normal arithmetic, and in some ways very different. Through practice, you’ll learn to become a finite field human calculator extraordinaire! To save your brain cells some pain, we’ll limit you to a finite field of size 4. What this means is that there are only four elements you’ll have to work with:

\[0, 1, A, \text{ and } B\]
You’ll be dealing with arithmetic operators that look like the ones you’re used to:

$$+, -, \times, \text{ and } \div$$

but they might act in strange ways! The strangeness is mostly due to the fact that A and B are NOT variables, so don’t think of them as such!!! Which of course makes figuring out expressions like ‘1+B’ all the more interesting...

The whole process should take between forty-five and sixty minutes. Do the best you can. It’s okay if you make mistakes. If you need to take a break before you’re finished, click on the ‘take a break’ link that appears after you’ve answered one of the questions.

Thanks in advance for your help.”

Each student will also receive a printed copy of the diagram in Figure A.1 to encourage students to take notes on what they have learned. The purpose of this “cheat sheet” is to reduce variance among students due to memory differences. We hope to focus on learning in the sense of understanding rather than solely on memorization skills.

After reading the instructions, students in all groups were given the same pre-test (see section A.4 below.) A subset of the students from group 1 was given the same pre-test a second time to discover whether students learn during the process of taking the pre-test.

Next the students were presented a training set of 24 problems to allow them to learn how to solve problems within this domain. The order of presentation of the 24 problems depended on which of the 3 groups a student has been assigned to (see section A.3 below for an explanation.)

Following the training set, all students were be given the same post-test (again, see section A.4.) Each student will see 54 problems in total which should be easily attainable within the allotted 1 hour.
A.3 Description of the Problems

The problem set is designed to teach 21 problems from the domain. The following table shows a list of the underlying problems being taught:

<table>
<thead>
<tr>
<th></th>
<th>primary</th>
<th>secondary</th>
<th>tertiary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1 + A</td>
<td>1 − B</td>
<td>A × A</td>
</tr>
<tr>
<td>2.</td>
<td>1 + B</td>
<td>B ÷ A</td>
<td>A − 1</td>
</tr>
<tr>
<td>3.</td>
<td>A + B</td>
<td>1 ÷ A</td>
<td>B × B</td>
</tr>
<tr>
<td>4.</td>
<td>A × B</td>
<td>1 − A</td>
<td>A ÷ B</td>
</tr>
<tr>
<td>5.</td>
<td>1 + 1</td>
<td>0 − B</td>
<td>1 ÷ B</td>
</tr>
<tr>
<td>6.</td>
<td>A + A</td>
<td>0 − 1</td>
<td>B − 1</td>
</tr>
<tr>
<td>7.</td>
<td>B + B</td>
<td>0 − A</td>
<td>B − A</td>
</tr>
</tbody>
</table>

The 7 problems in the “primary” column will be presented exactly as shown here. The other 14 problems will be embedded inside of composite problems and thus will need to be learned indirectly. Three extra problems derived from the first row will be added to the problems presented to students. The three new problems are designed to ensure that the student has a chance to verify that symmetry works...
in this domain for addition and multiplication. With the three new problems the initial table becomes:

<table>
<thead>
<tr>
<th>primary</th>
<th>secondary</th>
<th>tertiary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1 + A</td>
<td>1 - B</td>
</tr>
<tr>
<td>2.</td>
<td>A + 1</td>
<td>1 - B</td>
</tr>
<tr>
<td>3.</td>
<td>1 + B</td>
<td>B ÷ A</td>
</tr>
<tr>
<td>4.</td>
<td>A + B</td>
<td>1 ÷ A</td>
</tr>
<tr>
<td>5.</td>
<td>A × B</td>
<td>1 - A</td>
</tr>
<tr>
<td>6.</td>
<td>1 + 1</td>
<td>0 - B</td>
</tr>
<tr>
<td>7.</td>
<td>A + A</td>
<td>0 - 1</td>
</tr>
<tr>
<td>8.</td>
<td>B + B</td>
<td>0 - A</td>
</tr>
</tbody>
</table>

The following table shows the actual 24 problems that students will see:

<table>
<thead>
<tr>
<th>primary</th>
<th>secondary</th>
<th>tertiary</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1 + A</td>
<td>1 - (1 + A)</td>
</tr>
<tr>
<td>2.</td>
<td>A + 1</td>
<td>1 - (A + 1)</td>
</tr>
<tr>
<td>3.</td>
<td>1 + B</td>
<td>B ÷ (1 + B)</td>
</tr>
<tr>
<td>4.</td>
<td>A + B</td>
<td>(A + B) ÷ A</td>
</tr>
<tr>
<td>5.</td>
<td>A × B</td>
<td>(A × B) - A</td>
</tr>
<tr>
<td>6.</td>
<td>1 + 1</td>
<td>(1 + 1) - B</td>
</tr>
<tr>
<td>7.</td>
<td>A + A</td>
<td>(A + A) - 1</td>
</tr>
<tr>
<td>8.</td>
<td>B + B</td>
<td>(B + B) - A</td>
</tr>
</tbody>
</table>

As noted, the problems in the “primary” column will be presented as shown here. The problems in the “secondary” column require knowledge that may be gained by learning problems in the “primary” column. The problems in the “tertiary” column require knowledge that may be gained by learning those problems in the “secondary” column.

For example, row 1 in the first table shows that we would like to teach the
students how to solve the problems: $1 + A$, $1 - B$, and $A \times A$. Students will see an overt example of $1 + A$. However, they will not see an example of $1 - B$ alone, but rather, they will see $1 - (1 + A)$. If a student has previously learned that the solution to $1 + A$ is $B$ he should then be able to reduce $1 - (1 + A)$ to $1 - B$ allowing him to subsequently learn the solution to $1 - B$. Likewise, a student will not see the problem $A \times A$ but instead will see $(1 - B) \times A$. If the student has learned to solve $1 - B$ as $A$ from the “secondary” problem he should be able to reduce $(1 - B) \times A$ to $A \times A$. The other rows are constructed similarly. We will refer to such a sequence of three problems as a “problem chain”, or simply a “chain”.

A.3.1 Ordering

The order of presentation of the problems depends on the group a student is assigned to.

**Random Condition**

The 24 problems will be presented to the students in the random condition in random order without replacement.

**Expert Condition**

The 24 problems will be presented to students in the expert condition in the following fixed order:

1. $1 + A$
2. $1 - (1 + A)$
3. $(1 - B) \times A$
4. $A + 1$
5. $1 - (A + 1)$
6. $A \times (1 - B)$
7. $1 + B$
8. $B ÷ (1 + B)$
9. $(B ÷ A) - 1$
10. $A + B$
11. $(A + B) ÷ A$
12. $(1 ÷ A) × B$
13. $A × B$
14. $(A × B) - A$
15. $A ÷ (1 - A)$
16. $1 + 1$
17. $(1 + 1) - B$
18. $1 ÷ (0 - B)$
19. $A + A$
20. $(A + A) - 1$
21. $B - (0 - 1)$
22. $B + B$
23. $(B + B) - A$
24. $B - (0 - A)$

We assume that problem chains are most easily learned when each chain is presented in its proper order (primary, secondary, tertiary) before moving on to the next chain.
Learned Condition

The 24 problems will be presented to students in the learned condition in an order learned from the data collected from the random condition students.

A.4 Pre-test/Post-test

The pre-test and post-test are identical for all students. The purpose of the pre-test is to establish students’ level of prior knowledge. The purpose of the post-test is to analyze the differences in performance of students due to different training methods. The test contains problems of 5 types. “Trivial” problems are those we assume that the students already know due to their pre-existing understanding of algebra. “Primary”, “secondary”, and “tertiary” problems are problems that we expect to be new to students and that must be learn in the way outlined in section A.3. Finally, the “symmetry/composite” problems are designed to test whether the student has learned symmetry and to further test the students’ mastery of the core 24 problems.
<table>
<thead>
<tr>
<th></th>
<th>problem</th>
<th>purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$(0 + A) \div A$</td>
<td>trivial</td>
</tr>
<tr>
<td>2.</td>
<td>$(1 \times A) - 0$</td>
<td>trivial</td>
</tr>
<tr>
<td>3.</td>
<td>$(A \times 0) \div B$</td>
<td>trivial</td>
</tr>
<tr>
<td>4.</td>
<td>$B - (B \div 1)$</td>
<td>trivial</td>
</tr>
<tr>
<td>5.</td>
<td>$1 + 1$</td>
<td>primary</td>
</tr>
<tr>
<td>6.</td>
<td>$A + B$</td>
<td>primary</td>
</tr>
<tr>
<td>7.</td>
<td>$A \times (A + 1)$</td>
<td>primary</td>
</tr>
<tr>
<td>8.</td>
<td>$1 \div A$</td>
<td>secondary</td>
</tr>
<tr>
<td>9.</td>
<td>$B \div A$</td>
<td>secondary</td>
</tr>
<tr>
<td>10.</td>
<td>$B - 1$</td>
<td>tertiary</td>
</tr>
<tr>
<td>11.</td>
<td>$B - A$</td>
<td>tertiary</td>
</tr>
<tr>
<td>12.</td>
<td>$B + A$</td>
<td>symmetry/composite</td>
</tr>
<tr>
<td>13.</td>
<td>$(B + 1) \times (0 - A)$</td>
<td>symmetry/composite</td>
</tr>
<tr>
<td>14.</td>
<td>$(B \times A) \div (1 - A)$</td>
<td>symmetry/composite</td>
</tr>
<tr>
<td>15.</td>
<td>$(B + B) - (B - 1)$</td>
<td>symmetry/composite</td>
</tr>
</tbody>
</table>
APPENDIX B

BLAST: ADDITIONAL EXPERIMENT PROTOCOL DETAILS

This appendix contains further details about the protocol used in the BLAST experiments.

B.1 Verbal Instructions

Upon arriving in the lab, each subject was presented with a printed copy of the consent form (see Section C.3 of Appendix C) and was instructed to read and sign the form.

After signing the consent form, the following instructions were verbally given by the experimenter to each subject in the experiment, either individually, or as a group when running more than one subject:

“Please take a seat and log in. For a user name, do not use your own name. Just choose a silly word, something that can not be used to identify you later. After logging in, the system will give you instructions. After reading the instructions, the system will begin to teach you about the made-up language you read about in the experiment description. The system itself will decide when you are finished based on its estimate of how well you understand the language. The whole session will last no longer than 50 minutes, but it could be far shorter, so do the best you can.”

B.2 A Visual Guide

This section contains a sequence of screenshots of the BLAST system to document the process from the subject’s perspective.

The subject was initially presented with the screen seen in Figure B.1 requesting a new user name. This was followed by a request for a new password (Figure B.2).
Welcome to BLAST!
Prepare to have some fun learning a made up language!

Please enter a username.
Don’t use your real name, just make something up.

Figure B.1: Log In

Pick a password, Frodo.

Confirm your password:

Figure B.2: Enter Password
After the log-in process, subjects were given the brief set of instructions seen in Figures B.3 and B.4.

Instructions were followed by a request for an alertness level report, Figure B.5. Alertness reports were taken after every 5 learning events during training.

At this point training began, and the subject was presented with a sequence of multiple choice problems and hints. Figures B.6 and B.7 show an example of a multiple choice problem answered correctly. Figures B.8 and B.9 show an example of a problem answered incorrectly.

Figure B.10 shows an example of a hint.

Figures B.11 through B.13 show a problem sequence under a Choice condition.

Figures B.14 through B.17 show the start of a test, the first 2 questions, and the last question.

Figure B.18 shows the feedback message for a low test score.

Figure B.19 shows the feedback message for a high test score which enables a
I’ll also occasionally ask you how you are feeling (whether you are frustrated, happy, interested, etc.) These questions are to help me understand how well I am doing as a teacher. Your answers will allow me to adjust my teaching style to match your needs.

I hope you enjoy it!

Figure B.4: Instructions II

Figure B.5: Alertness Report
Figure B.6: First Multiple Choice Problem

Figure B.7: Correct Answer
Figure B.8: Second Multiple Choice Problem

Figure B.9: Wrong Answer
Figure B.10: Hint

Figure B.11: Choice Condition: Selecting a Problem
Figure B.12: Choice Condition: Problem Selected

Figure B.13: Choice Condition: Correct Answer
Figure B.14: Test Intro

Figure B.15: Test Question 1
Figure B.16: Test Question 2
Figure B.17: Test Question 20

Figure B.18: Test Finished, Continue
student to leave the experiment early with a feeling of pride for having mastered the Muq-Duq language.
Figure B.19: Test Finished, Achieved Mastery!

Figure B.20: Bye
APPENDIX C

IRB DOCUMENTS

C.1 CITI Completion Report
C.2 IRB 2009
C.3 IRB 2010
CITI Collaborative Institutional Training Initiative

Human Research Curriculum Completion Report
Printed on

Learner: Derek Green (username: dtgreen)
Institution: University of Arizona
Contact Information: Computer Science
Coudé-Simpson Bldg Rm 721
1040 E. 4th Street
Tucson, AZ 85721 United States
Department: Computer Science
Email: dtgreen@cs.arizona.edu

Social & Behavioral Research Investigators:

Stage: Basic Course Passed on 06/22/09 (Ref # 2571408)

<table>
<thead>
<tr>
<th>Required Modules</th>
<th>Date Completed</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>02/23/09</td>
<td>no quiz</td>
</tr>
<tr>
<td>Students in Research - SBR</td>
<td>02/23/09</td>
<td>5/5 (100%)</td>
</tr>
<tr>
<td>History and Ethical Principles - SBR</td>
<td>05/25/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>Defining Research with Human Subjects - SBR</td>
<td>05/26/09</td>
<td>5/5 (100%)</td>
</tr>
<tr>
<td>The Regulations and The Social and Behavioral Sciences - SBR</td>
<td>05/26/09</td>
<td>5/5 (100%)</td>
</tr>
<tr>
<td>Assessing Risk in Social and Behavioral Sciences - SBR</td>
<td>05/26/09</td>
<td>5/5 (100%)</td>
</tr>
<tr>
<td>Informed Consent - SBR</td>
<td>06/15/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>Privacy and Confidentiality - SBR</td>
<td>06/15/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>Research with Children - SBR</td>
<td>06/19/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>Research in Public Elementary and Secondary Schools - SBR</td>
<td>06/19/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>International Research - SBR</td>
<td>06/19/09</td>
<td>3/3 (100%)</td>
</tr>
<tr>
<td>Internet Research - SBR</td>
<td>06/19/09</td>
<td>5/5 (100%)</td>
</tr>
<tr>
<td>HIPAA and Human Subjects Research</td>
<td>06/19/09</td>
<td>2/2 (100%)</td>
</tr>
<tr>
<td>Workers as Research Subjects: A Vulnerable Population</td>
<td>06/19/09</td>
<td>4/4 (100%)</td>
</tr>
<tr>
<td>Conflicts of Interest in Research Involving Human Subjects</td>
<td>06/19/09</td>
<td>2/2 (100%)</td>
</tr>
<tr>
<td>University of Arizona</td>
<td>06/22/09</td>
<td>no quiz</td>
</tr>
<tr>
<td>UA - Native American Module</td>
<td>06/19/09</td>
<td>5/5 (100%)</td>
</tr>
</tbody>
</table>

For this Completion Report to be valid, the learner listed above must be affiliated with a CITI participating institution. Falsified information and data will be disregarded.

Figure C.1: CITICompletionReport
<table>
<thead>
<tr>
<th>Date:</th>
<th>10/01/09</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investigator:</td>
<td>Derek Green, PhD Candidate</td>
</tr>
<tr>
<td>Department:</td>
<td>Comp Sci</td>
</tr>
<tr>
<td>Advisor:</td>
<td>Paul Cohen, PhD</td>
</tr>
<tr>
<td>Project No/Title:</td>
<td>09-0876-02 Improving Computer-Based Teaching By Using Data From Prior Students to Train Problem Selection Algorithms</td>
</tr>
<tr>
<td>Current Period of Approval:</td>
<td>10/01/09 – 09/30/10</td>
</tr>
</tbody>
</table>

**IRB Committee Information**

<table>
<thead>
<tr>
<th>IRB2 – IRB00001751</th>
<th>Expedited Review – 10/01/09</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWA Number:</td>
<td>FWA00004218</td>
</tr>
<tr>
<td>Nature of Submission:</td>
<td>New Project</td>
</tr>
</tbody>
</table>

**Documents Reviewed Concurrently**

<table>
<thead>
<tr>
<th>Description</th>
<th>Review:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Review Form (09/04/09)</td>
<td>Appr</td>
</tr>
<tr>
<td>Subject Consent Form [v. 10/01/09]</td>
<td>Appr</td>
</tr>
<tr>
<td>TQMT</td>
<td>Appr</td>
</tr>
<tr>
<td>Questionnaires/Surveys</td>
<td>Math questionnaire</td>
</tr>
<tr>
<td>Grant Application</td>
<td>DARPA BAA submitted 6/13/08</td>
</tr>
</tbody>
</table>

**Committee/Chair Determination**

- **Expedite Approval (45 CFR 46.110 Category 6):** Collection of data from voice, video, digital, or image recordings made for research purposes.

- **Expedite Approval (45 CFR 46.110 Category 7):** Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and social behavior). Special types of research that involve no more than minimal risk and meet one of the following four criteria: (a) The research involves contact only with human participants of any age or setting; (b) The research involves surveys or interview studies; (c) The research involves observational data collection; (d) The research involves medical care, patient management, or data collection in the context of providing health or other benefits. No more than minimal risk is determined to be present for research involving child's participation if the research involves medical care, patient management, or data collection in the context of providing health or other benefits.

**Committee/Chair Determination**

- Approved as submitted effective 10/01/09

**Additional Determinations:**

- Expedite Approval (45 CFR 46.110 Category 6): Collection of data from voice, video, digital, or image recordings made for research purposes.

- Expedite Approval (45 CFR 46.110 Category 7): Research on individual or group characteristics or behavior (including, but not limited to, research on perception, cognition, motivation, identity, language, communication, cultural beliefs or practices, and social behavior) or research employing survey, interview, oral history, focus group, program evaluation, human factors evaluation, or quality assurance methodologies.

Elaine G. Jones, PhD  
Chair, IRB 2 Committee  
UA Institutional Review Board  
cc: Departmental/College Review Committee

10-01-09  
Chair, IRB 2 Committee  
UA Institutional Review Board

Form version: 09/23/09

Figure C.2: IRB Approval 2009
Informed Consent

Improving computer-based teaching by using data from prior students to train problem selection algorithms

Introduction

You are being invited to take part in a research study. The information in this form is provided to help you decide whether or not to take part. Study personnel will be available to answer your questions and provide additional information. If you decide to take part in the study, you will be asked to sign this consent form. A copy of this form will be given to you.

What is the purpose of this research study?

We want to train a computer algorithm with data from one set of students to estimate the best way to present materials to students in the future. Our overall goal is to improve curriculum development for electronic educational materials, such as digital libraries. In contrast to print textbooks, most electronic resources are not linked or related so that learners would know which to study first and what to leave until basic concepts have been mastered. We hope to find ways to have a computer use data from past student users to create an organized curriculum from the materials, and then continually improve the curriculum as more students use the electronic resources.

Why are you being asked to participate?

You are being invited because we would like to obtain data from sets of students who will work with unfamiliar electronic materials to use to train and test the performance of our computer program. We are looking for participants who are able to use a computer and website, and who can understand and read English.

How many people will be asked to participate in this study?

Approximately 48 persons will be asked to participate in this study.

What will happen during this study?

You will be asked to complete a pre-test of your knowledge of a type of puzzle in which mathematical operations are performed on a small set of symbols. The pre-test includes 15 simple examples. You are not expected to know how to solve the puzzles yet.

Version: 10-01-09

Figure C.3: Consent Form 2009, Page 1
You will then be presented a series of examples of such puzzles to help you learn how to solve them. After each problem, you will be asked if you would like to take a short break, continue, or stop all together. After every five questions you will be asked to rate your level of alertness and interest. You can skip any problem that you want to. There are 100 problems. You will work on this for no more than 40 minutes. You do not have to complete all of the problems.

Finally, you will be asked to complete a short post-test with 10 new problems.

How long will I be in this study?

One 60 minute session will be needed to complete this study.

Are there any risks to me?

The things that you will be doing have very little risk. You might occasionally find some of the problems frustrating, but you can skip any problems that you don’t want to complete, and you can stop or take a break whenever you want.

Are there any benefits to me?

The only benefit that you would receive is 1 hour of experimental psychology credit for participating in the study.

What are the alternatives for participating in this study?

The alternative is to sign up for another study or to choose to complete another assignment as described in your syllabus.

Will there be any costs to me?

Aside from your time, there are no costs for being in the study.

Will I be paid to participate in the study?

You will not be paid for your participation.

Will video or audio recordings be made of me during the study?

No.

Will the information that is obtained from me be kept confidential?

The only persons who will know that you participated in this study will be the research team members: Derek Green, Dr. Carole Beal, and Tasneem Kaochar.
Your problem solving on the computer will not be associated with your name or any identifying information. You will not be identified in any reports or publications resulting from the study.

It is possible that representatives of the sponsor that supports the research study will want to come to The University of Arizona to review the study information. Representatives of regulatory agencies including The University of Arizona Human Subjects Protection Program may access the study records. However, there is no information in the records that would allow you to be identified.

What if I am harmed by the study procedures?
The risks of participation are minimal in this study.

May I change my mind about participating?
Your participation in this study is voluntary. You may decide to not begin or to stop the study at any time without any penalty. You can tell the researcher directing the session that you have decided to stop and he or she will help you log off the computer.

Whom can I contact for additional information?
You can call the Principal Investigator to tell him/her about a concern or complaint about this research study. The Principal Investigator Derek Green PhD Candidate can be called at (520)626-7443. If you have questions about your rights as a research subject you may call the University of Arizona Human Subjects Protection Program office at (520) 626-6721. If you have questions, complaints, or concerns about the research and cannot reach the Principal Investigator or want to talk to someone other than the Investigator, you may call the University of Arizona Human Subjects Protection Program office. (If out of state use the toll-free number 1-866-278-1455.) If you would like to contact the Human Subjects Protection Program via the web (this can be anonymous), please visit http://www.irb.arizona.edu/contact/

Your Signature
By signing this form, I affirm that I have read the information contained in the form, that the study has been explained to me, that my questions have been answered and that I agree to take part in this study. I do not give up any of my legal rights by signing this form.

__________________________________
Name (Printed)

__________________________________   ______________

Version: 10-01-09

Page 3 of 4  Participant’s Initials____
Statement by person obtaining consent

I certify that I have explained the research study to the person who has agreed to participate, and that he or she has been informed of the purpose, the procedures, the possible risks and potential benefits associated with participation in this study. Any questions raised have been answered to the participant’s satisfaction.

Name of study personnel

Study personnel Signature  Date signed
**HSPP Correspondence Form**

**Date:** 09/01/10  
**Investigator:** Derek Green, Ph.D. Candidate  
**Advisor:** Paul Cohen, Ph.D.  
**Department:** Computer Science  
**Project No./Title:** 09-0876-02 Improving Computer-Based Teaching By Using Data From Prior Students to Train Problem Selection Algorithms  
**Current Period of Approval:** 09/30/10 – 09/29/11

**IRB Committee Information:**

<table>
<thead>
<tr>
<th>Document Reviewed Concurrently</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuing Review Report (received 08/03/10)</td>
<td>Appr</td>
</tr>
<tr>
<td>Informed Consent (version 08/23/10)</td>
<td>Appr</td>
</tr>
<tr>
<td>VOTF (received 08/23/10)</td>
<td>Appr</td>
</tr>
</tbody>
</table>

**Description of Modifications:**

- **Revised Informed Consent form** (removing the HSPP 1-800 number); personnel changes (adding Cohen; removing Kaochar).

**Determination:**

- Approved as submitted effective 09/01/10

**Comments:**

- Continuing Review Category Status – Enrollment in Progress or Still Planned

**Regulatory Determinations:**

- Criteria for Approval has been met (45 CFR 46.111)
- Expedite Approval (45 CFR 46.110 Category 6)
- Expedite Approval (45 CFR 46.110 Category 7)

---

**Elaine G. Jones, RN, PhD**  
Co-Chair, IRB 2 Committee  
UA Institutional Review Board

Date: 09/01/10

---

**Figure C.7: IRB Approval 2010**
Form version: 10/01/2010

HSPP Correspondence Form

Date: 11/08/10

Investigator: Derek Green, PhD Candidate
Department: Computer Science
Advisor: Paul Cohen, PhD

Project No/Title: 09-0876-02 Improving Computer-Based Teaching By Using Data From Prior Students to Train Problem Selection Algorithms

Current Period of Approval: 09/30/10 - 09/29/11
Select the “FORM: Continuing Review Progress Report” no later than 45 days prior to the end of the approval period listed above.

To: IRB2 – IRB00001751
FWA Number: FWA00004218
Expedited Review – Amendment

Documents Reviewed Concurrently

Appr

Request for Amendment Form – PI Initiated Changes (dated 10/29/10)
Consenting Instruments
Informed Consent (10/29/10)
Data Collection Instruments: Change the form of the puzzle used

Description of Submission

Revised study documents (The form of the puzzle being used to collect data from participants is being changed to use a made up language to describe simple geometric shapes. This language the domain has been constructed to make it easier for the algorithms to track the progress of subjects as they learn about the made-up language)

Determination

Approved as submitted effective 11/08/10

Elaine G. Jones, PhD, RN
Chair, IRB 2 Committee
UA Institutional Review Board

Figure C.8: IRB Amendment 2010
Informed Consent

Improving computer-based teaching by using data from prior students to train problem selection algorithms

Introduction

You are being invited to take part in a research study. The information in this form is provided to help you decide whether or not to take part. Study personnel will be available to answer your questions and provide additional information. If you decide to take part in the study, you will be asked to sign this consent form. A copy of this form will be given to you.

What is the purpose of this research study?

We want to train a computer algorithm with data from one set of students to estimate the best way to present materials to students in the future. Our overall goal is to improve curriculum development for electronic educational materials, such as digital libraries. In contrast to print textbooks, most electronic resources are not linked or related so that learners would know which to study first and what to leave until basic concepts have been mastered.

We hope to find ways to have a computer use data from past student users to create an organized curriculum from the materials, and then continually improve the curriculum as more students use the electronic resources.

Why are you being asked to participate?

You are being invited because we would like to obtain data from sets of students who will work with unfamiliar electronic materials to use to train and test the performance of our computer program. We are looking for participants who are able to use a computer and website, and who can understand and read English.

How many people will be asked to participate in this study?

Approximately 100 persons will be asked to participate in this study.

What will happen during this study?

During this study you will learn how to use words from a made-up language to describe simple images of geometric shapes. You will be presented a series of questions and statements about the language to help you learn how to use it. You may also be asked to rank your confidence in an answer you give, your level of frustration, or your current level of interest. You can skip any problem that you...
want to. You will work on this for no longer than 50 minutes. You do not have to complete all of the problems.

How long will I be in this study?

One 60 minute session will be needed to complete this study.

Are there any risks to me?

The things that you will be doing have very little risk. You might occasionally find some of the puzzles frustrating, but you can skip any problems that you don't want to complete, and you can stop or take a break whenever you want.

Are there any benefits to me?

The only benefit that you would receive is 1 hour of experimental psychology credit for participating in the study.

What are the alternatives for participating in this study?

The alternative is to sign up for another study or to choose to complete another assignment as described in your syllabus.

Will there be any costs to me?

Aside from your time, there are no costs for being in the study.

Will I be paid to participate in the study?

You will not be paid for your participation.

Will video or audio recordings be made of me during the study?

No.

Will the information that is obtained from me be kept confidential?

The only persons who will know that you participated in this study will be the research team members: Derek Green, and Dr. Carole Beal.

Your problem solving on the computer will not be associated with your name or any identifying information. You will not be identified in any reports or publications resulting from the study.

It is possible that representatives of the sponsor that supports the research study will want to come to The University of Arizona to review the study information.
Representatives of regulatory agencies including The University of Arizona Human Subjects Protection Program may access the study records. However, there is no information in the records that would allow you to be identified.

What if I am harmed by the study procedures?

The risks of participation are minimal in this study.

May I change my mind about participating?

Your participation in this study is voluntary. You may decide to not begin or to stop the study at any time without any penalty. You can tell the researcher directing the session that you have decided to stop and he or she will help you log off the computer.

Whom can I contact for additional information?

You can call the Principal Investigator to tell him/her about a concern or complaint about this research study. The Principal Investigator Derek Green PhD Candidate can be called at (520)626-7443. If you have questions about your rights as a research subject you may call the University of Arizona Human Subjects Protection Program office at (520) 626-7443. If you have questions, complaints, or concerns about the research and cannot reach the Principal Investigator or want to talk to someone other than the Investigator, you may call the University of Arizona Human Subjects Protection Program office. If you would like to contact the Human Subjects Protection Program via the web (this can be anonymous), please visit http://orcr.vpr.arizona.edu/irb/contact/.
Your Signature

By signing this form, I affirm that I have read the information contained in the form, that the study has been explained to me, that my questions have been answered and that I agree to take part in this study. I do not give up any of my legal rights by signing this form.

Name (Printed) ___________________ Date signed __________

Participant’s Signature ___________________ Date signed __________

Statement by person obtaining consent

I certify that I have explained the research study to the person who has agreed to participate, and that he or she has been informed of the purpose, the procedures, the possible risks and potential benefits associated with participation in this study. Any questions raised have been answered to the participant’s satisfaction.

Name of study personnel ___________________ Date signed __________

Study personnel Signature ___________________ Date signed __________

Figure C.12: Consent Form 2010, Page 4
REFERENCES


REFERENCES – Continued


