

TABLE OF CONTENTS

Chapter 1	1
<i>Gives the student a "feel" for what it is like to program in Prolog. Introduces objects, relationships, facts, rules, variables.</i>	
1.1 Facts	2
1.2 Questions	5
1.3 Variables	6
1.4 Conjunctions	8
1.5 Rules	13
1.6 Summary and Exercises	19
Chapter 2	21
<i>More detailed presentation of Prolog syntax and data structures.</i>	
2.1 Syntax	21
2.2 Characters	25
2.3 Operators	26
2.4 Equality and Matching	28
2.5 Arithmetic	30
2.6 Summary of Satisfying Goals	34
Chapter 3	43
<i>Representing objects and relationships by using "trees" and "lists". Developing several standard Prolog programming techniques.</i>	
3.1 Structures and Trees	43
3.2 Lists	45
3.3 Recursive Search	49
3.4 Mapping	53
3.5 Recursive Comparison	56
3.6 Joining Structures Together	59
3.7 Accumulators	62
3.8 Difference Structures	66
Chapter 4	69
<i>How a set of clauses generates a set of solutions. Using "cut" to modify the control sequence of running Prolog programs.</i>	
4.1 Generating Multiple Solutions	70
4.2 The "Cut"	75
4.3 Common Uses of the Cut	79
4.4 Problems with the Cut	91

Chapter 5	93
<i>Facilities available for the input and output of characters and structures. Developing a program to read sentences from the user and represent the sentence as a list of words, which can be used with the Grammar Rules of Chapter 9.</i>	
5.1 Reading and Writing Terms	95
5.2 Reading and Writing Characters	98
5.3 Reading English Sentences	100
5.4 Reading and Writing Files	103
5.5 Declaring Operators	106
Chapter 6	109
<i>Definition of the "core" built-in predicates, with sensible examples of how each one is used. By this point, the reader should be able to read reasonably complex programs, and should therefore be able to absorb the built-in predicates by seeing them in use.</i>	
6.1 Entering New Clauses	109
6.2 Success and Failure	111
6.3 Classifying Terms	112
6.4 Treating Clauses as Terms	113
6.5 Constructing and Accessing Components of Structures	117
6.6 Affecting Backtracking	121
6.7 Constructing Compound Goals	123
6.8 Equality	126
6.9 Input and Output	127
6.10 Handling Files	129
6.11 Evaluating Arithmetic Expressions	130
6.12 Comparing Numbers	131
6.13 Watching Prolog at Work	131
Chapter 7	133
<i>Many example programs are given, covering a wide range of interests. New examples include list processing, set operations, symbolic differentiation and simplification of formulae.</i>	
7.1 A Sorted Tree Dictionary	133
7.2 Searching a Maze	136
7.3 The Towers of Hanoi	140
7.4 Parts Inventory	141
7.5 List Processing	143
7.6 Representing and Manipulating Sets	147
7.7 Sorting	149
7.8 Using the Database: random, gensym, findall	152

7.9 Searching Graphs	158
7.10 Sift the Two's and Sift the Three's	164
7.11 Symbolic Differentiation	165
7.12 Mapping Structures and Transforming Trees	167
7.13 Manipulating Programs	170
Chapter 8	175
<i>By this point, the reader will be able to write reasonable programs, and so the problem of debugging will be relevant. Flow of control model, hints about common bugs, techniques of debugging.</i>	
8.1 Laying out Programs	176
8.2 Common Errors	179
8.3 The Tracing Model	182
8.4 Tracing and Spy Points	188
8.5 Fixing Bugs	198
Chapter 9	201
<i>Applications of existing techniques. Using Grammar Rules. Examining the design decisions for some aspects of analysing natural language with Grammar Rules.</i>	
9.1 The Parsing Problem	201
9.2 Representing the Parsing Problem in Prolog	204
9.3 The Grammar Rule Notation	209
9.4 Adding Extra Arguments	211
9.5 Adding Extra Tests	215
9.6 Summary	217
Chapter 10	221
<i>Predicate Calculus, clausal form, resolution theorem proving, logic programming.</i>	
10.1 Brief Introduction to Predicate Calculus	221
10.2 Clausal Form	224
10.3 A Notation for Clauses	230
10.4 Resolution and Proving Theorems	232
10.5 Horn Clauses	236
10.6 Prolog	237
10.7 Prolog and Logic Programming	239
Chapter 11	243
<i>A selection of suggested exercises, projects, and problems.</i>	
11.1 Easier Projects	243
11.2 Advanced Projects	246

APPENDICES

A. Answers to Selected Exercises	249
B. Clausal Form Program Listings	254
C. Different Versions of Prolog	260
D. Edinburgh Prolog	264
E. micro-Prolog	273
INDEX	277