

Table of Contents

1	Tutorial Introduction	1
	<i>Gives the student a feel for what it is like to program in Prolog.</i>	
	<i>Introduces objects, relationships, facts, rules, variables</i>	
1.1	Prolog	1
1.2	Objects and Relationships	2
1.3	Programming	3
1.4	Facts	4
1.5	Questions	6
1.6	Variables	8
1.7	Conjunctions	10
1.8	Rules	16
1.9	Summary and Exercises	22
2	A Closer Look	25
	<i>More detailed presentation of Prolog syntax and data structures</i>	
2.1	Syntax	25
2.1.1	Constants	26
2.1.2	Variables	27
2.1.3	Structures	27
2.2	Characters	29
2.3	Operators	30
2.4	Equality and Unification	32
2.5	Arithmetic	34
2.6	Summary of Satisfying Goals	37
2.6.1	Successful satisfaction of a conjunction of goals	38

2.6.2	Consideration of goals in backtracking	42
2.6.3	Unification.....	43
<hr/>		
3	Using Data Structures	47
	<i>Representing objects and relationships by using trees and lists.</i>	
	<i>Developing several standard Prolog programming techniques</i>	
3.1	Structures and Trees	47
3.2	Lists	50
3.3	Recursive Search	53
3.4	Mapping	57
3.5	Recursive Comparison	60
3.6	Joining Structures Together	63
3.7	Accumulators	67
3.8	Difference Structures	70
4	Backtracking and the “Cut”	73
	<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	74
4.2	The “Cut”	80
4.3	Common Uses of the Cut	85
4.3.1	Confirming the Choice of a Rule	85
4.3.2	The “cut-fail” Combination	90
4.3.3	Terminating a “generate and test”	92
4.4	Problems with the Cut	96
5	Input and Output	99
	<i>Facilities available for the input and output of characters and structures.</i>	
	<i>Developing a program to read sentences from the user and represent the structure as a list of words, which can be used with the Grammar Rules of Chapter 9</i>	
5.1	Reading and Writing Terms	100
5.1.1	Reading Terms	100
5.1.2	Writing Terms	101
5.2	Reading and Writing Characters	104
5.2.1	Reading Characters	105
5.2.2	Writing Characters	106

5.3	Reading English Sentences	108
5.4	Reading and Writing Files	111
5.4.1	Opening and closing streams	112
5.4.2	Changing the current input and output	113
5.4.3	Consulting	115
5.5	Declaring Operators	116
6	Built-in Predicates	119
	<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	120
6.2	Success and Failure	121
6.3	Classifying Terms	122
6.4	Treating Clauses as Terms	123
6.5	Constructing and Accessing Components of Structures	127
6.6	Affecting Backtracking	132
6.7	Constructing Compound Goals	134
6.8	Equality	136
6.9	Input and Output	137
6.10	Handling Files	138
6.11	Evaluating Arithmetic Expressions	139
6.12	Comparing Terms	140
6.13	Watching Prolog at Work	143
7	More Example Programs	145
	<i>Many example programs are given, covering a wide range of interests. Examples include list processing, set operations, symbolic differentiation and simplification of formulae</i>	
7.1	A Sorted Tree Dictionary	145
7.2	Searching a Maze	148
7.3	The Towers of Hanoi	152
7.4	Parts Inventory	153
7.5	List Processing	155
7.6	Representing and Manipulating Sets	159
7.7	Sorting	161
7.8	Using the Database	164

7.8.1	Random	164
7.8.2	Gensym	165
7.8.3	Findall	167
7.9	Searching Graphs	169
7.10	Sift the Two's and Sift the Three's	174
7.11	Symbolic Differentiation	177
7.12	Mapping Structures and Transforming Trees	179
7.13	Manipulating Programs	182
7.14	Bibliographic Notes	185

8 Debugging Prolog Programs 187

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.

8.1	Laying out Programs	188
8.2	Common Errors	191
8.3	The Tracing Model	194
8.4	Tracing and Spy Points	200
8.4.1	Examining the Goal	204
8.4.2	Examining the Ancestors	205
8.4.3	Altering the Degree of Tracing	206
8.4.4	Altering the Satisfaction of the Goal	207
8.4.5	Other Options	209
8.4.6	Summary	209
8.5	Fixing Bugs	210

9 Using Prolog Grammar Rules 213

Applications of existing techniques. Using Grammar Rules.

Examining the design decisions for some aspects of analysing natural language with Grammar Rules

9.1	The Parsing Problem	213
9.2	Representing the Parsing Problem in Prolog	216
9.3	The Grammar Rule Notation	221
9.4	Adding Extra Arguments	223
9.5	Adding Extra Tests	227
9.6	Summary	230
9.7	Translating Language into Logic	231
9.8	More General Use of Grammar Rules	233

10 The Relation of Prolog to Logic	237
<i>Predicate Calculus, clausal form, resolution theorem proving, logic programming</i>	
10.1 Brief Introduction to Predicate Calculus	237
10.2 Clausal Form	240
10.3 A Notation for Clauses	246
10.4 Resolution and Proving Theorems	248
10.5 Horn Clauses	251
10.6 Prolog	252
10.7 Prolog and Logic Programming	254
11 Projects in Prolog	259
<i>A selection of suggested exercises, projects and problems</i>	
11.1 Easier Projects	259
11.2 Advanced Projects	262
A Answers to Selected Exercises	267
B Clausal Form Program Listings	271
C Writing Portable Standard Prolog Programs	277
<i>The Prolog standard, writing portable programs and dealing with different Prolog implementations</i>	
C.1 Standard Prolog for Portability	277
C.2 Different Prolog Implementations	278
C.3 Issues to Look Out For	279
C.4 Definitions of some Standard Predicates	280
C.4.1 Character Processing	281
C.4.2 Directives	283
C.4.3 Stream Input/Output	284
C.4.4 Miscellaneous	287
D Code to Support DCGs	289
D.1 DCG Support Code	290
Index	295