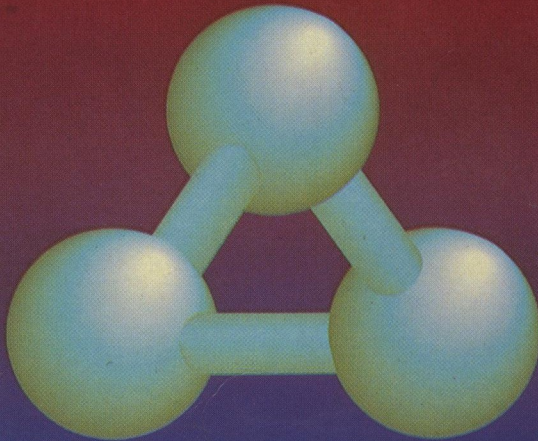




INTERNATIONAL  
COMPUTER SCIENCE  
SERIES

# PROLOG PROGRAMMING FOR ARTIFICIAL INTELLIGENCE

Ivan Bratko



# CONTENTS

<b>Foreword</b>		<b>vii</b>
<b>Preface</b>		<b>xi</b>
<b>PART ONE THE PROLOG LANGUAGE</b>		<b>1</b>
<b>Chapter 1 An Overview of Prolog</b>		<b>3</b>
1.1 An example program: defining family relations		3
1.2 Extending the example program by rules		8
1.3 A recursive rule definition		14
1.4 How Prolog answers questions		19
1.5 Declarative and procedural meaning of programs		24
<b>Chapter 2 Syntax and Meaning of Prolog Programs</b>		<b>27</b>
2.1 Data objects		27
2.2 Matching		35
2.3 Declarative meaning of Prolog programs		40
2.4 Procedural meaning		43
2.5 Example: monkey and banana		49
2.6 Order of clauses and goals		53
2.7 Remarks on the relation between Prolog and logic		60
<b>Chapter 3 Lists, Operators, Arithmetic</b>		<b>64</b>
3.1 Representation of lists		64
3.2 Some operations on lists		67
3.3 Operator notation		78
3.4 Arithmetic		84
<b>Chapter 4 Using Structures: Example Programs</b>		<b>93</b>
4.1 Retrieving structured information from a database		93
4.2 Doing data abstraction		97
4.3 Simulating a non-deterministic automaton		99
4.4 Travel planning		103
4.5 The eight queens problem		108

<b>Chapter 5</b>	<b>Controlling Backtracking</b>	120
5.1	Preventing backtracking	120
5.2	Examples using cut	125
5.3	Negation as failure	129
5.4	Problems with cut and negation	133
<b>Chapter 6</b>	<b>Input and Output</b>	137
6.1	Communication with files	137
6.2	Processing files of terms	140
6.3	Manipulating characters	147
6.4	Constructing and decomposing atoms	149
6.5	Reading programs: <b>consult</b> , <b>reconsult</b>	152
<b>Chapter 7</b>	<b>More Built-in Procedures</b>	155
7.1	Testing the type of terms	155
7.2	Constructing and decomposing terms: <b>=..</b> , <b>arg</b> , <b>name</b>	163
7.3	Various kinds of equality	168
7.4	Database manipulation	169
7.5	Control facilities	174
7.6	<b>bagof</b> , <b>setof</b> and <b>findall</b>	175
<b>Chapter 8</b>	<b>Programming Style and Technique</b>	179
8.1	General principles of good programming	179
8.2	How to think about Prolog programs	181
8.3	Programming style	184
8.4	Debugging	187
8.5	Efficiency	188
<b>PART TWO</b>	<b>PROLOG IN ARTIFICIAL INTELLIGENCE</b>	201
<b>Chapter 9</b>	<b>Operations on Data Structures</b>	203
9.1	Representing and sorting lists	203
9.2	Representing sets by binary trees	211
9.3	Insertion and deletion in a binary dictionary	217
9.4	Displaying trees	222
9.5	Graphs	224
<b>Chapter 10</b>	<b>Advanced Tree Representations</b>	233
10.1	The 2-3 dictionary	233
10.2	AVL-tree: an approximately balanced tree	241
<b>Chapter 11</b>	<b>Basic Problem-Solving Strategies</b>	246
11.1	Introductory concepts and examples	246
11.2	Depth-first search strategy	251
11.3	Breadth-first search strategy	256

11.4	Comments on searching graphs, on optimality, and on search complexity	262
<b>Chapter 12</b>	<b>Best First: A Heuristic Search Principle</b>	265
12.1	Best-first search	265
12.2	Best-first search applied to the eight puzzle	273
12.3	Best-first search applied to scheduling	278
<b>Chapter 13</b>	<b>Problem Reduction and AND/OR Graphs</b>	286
13.1	AND/OR graph representation of problems	286
13.2	Examples of AND/OR representation	290
13.3	Basic AND/OR search procedures	294
13.4	Best-first AND/OR search	299
<b>Chapter 14</b>	<b>Expert Systems</b>	314
14.1	Functions of an expert system	314
14.2	Main structure of an expert system	315
14.3	<i>If-then</i> rules for representing knowledge	316
14.4	Developing the shell	323
14.5	Implementation	329
14.6	Dealing with uncertainty	347
14.7	Concluding remarks	355
<b>Chapter 15</b>	<b>Game Playing</b>	359
15.1	Two-person, perfect-information games	359
15.2	The minimax principle	361
15.3	The alpha-beta algorithm: an efficient implementation of minimax	364
15.4	Minimax-based programs: refinements and limitations	368
15.5	Pattern knowledge and the mechanism of 'advice'	370
15.6	A chess endgame program in Advice Language 0	373
<b>Chapter 16</b>	<b>Pattern-Directed Programming</b>	390
16.1	Pattern-directed architecture	390
16.2	A simple interpreter for pattern-directed programs	394
16.3	A simple theorem prover	396
16.4	Concluding remarks	402
<b>Solutions to Selected Exercises</b>		405
<b>Index</b>		419