

# Contents

<b>Figures</b>	<b>xiii</b>
<b>Programs</b>	<b>xvii</b>
<b>Series Foreword</b>	<b>xxv</b>
<b>Foreword</b>	<b>xxvii</b>
<b>Preface</b>	<b>xxx</b>
<b>Preface to First Edition</b>	<b>xxxv</b>
<b>Introduction</b>	<b>1</b>

## ***I Logic Programs*** 9

<b>1</b>	<b>Basic Constructs</b>	<b>11</b>
1.1	Facts	11
1.2	Queries	12
1.3	The Logical Variable, Substitutions, and Instances	13
1.4	Existential Queries	14
1.5	Universal Facts	15
1.6	Conjunctive Queries and Shared Variables	16
1.7	Rules	18

1.8	A Simple Abstract Interpreter	22
1.9	The Meaning of a Logic Program	25
1.10	Summary	27
<b>2</b>	<b>Database Programming</b>	<b>29</b>
2.1	Simple Databases	29
2.2	Structured Data and Data Abstraction	35
2.3	Recursive Rules	39
2.4	Logic Programs and the Relational Database Model	42
2.5	Background	44
<b>3</b>	<b>Recursive Programming</b>	<b>45</b>
3.1	Arithmetic	45
3.2	Lists	56
3.3	Composing Recursive Programs	65
3.4	Binary Trees	72
3.5	Manipulating Symbolic Expressions	78
3.6	Background	84
<b>4</b>	<b>The Computation Model of Logic Programs</b>	<b>87</b>
4.1	Unification	87
4.2	An Abstract Interpreter for Logic Programs	91
4.3	Background	98
<b>5</b>	<b>Theory of Logic Programs</b>	<b>101</b>
5.1	Semantics	101
5.2	Program Correctness	105
5.3	Complexity	108
5.4	Search Trees	110
5.5	Negation in Logic Programming	113
5.6	Background	115

**II The Prolog Language 117**

- 6 Pure Prolog 119**
  - 6.1 The Execution Model of Prolog 119
  - 6.2 Comparison to Conventional Programming Languages 124
  - 6.3 Background 127
  
- 7 Programming in Pure Prolog 129**
  - 7.1 Rule Order 129
  - 7.2 Termination 131
  - 7.3 Goal Order 133
  - 7.4 Redundant Solutions 136
  - 7.5 Recursive Programming in Pure Prolog 139
  - 7.6 Background 147
  
- 8 Arithmetic 149**
  - 8.1 System Predicates for Arithmetic 149
  - 8.2 Arithmetic Logic Programs Revisited 152
  - 8.3 Transforming Recursion into Iteration 154
  - 8.4 Background 162
  
- 9 Structure Inspection 163**
  - 9.1 Type Predicates 163
  - 9.2 Accessing Compound Terms 167
  - 9.3 Background 174
  
- 10 Meta-Logical Predicates 175**
  - 10.1 Meta-Logical Type Predicates 176
  - 10.2 Comparing Nonground Terms 180
  - 10.3 Variables as Objects 182
  - 10.4 The Meta-Variable Facility 185
  - 10.5 Background 186

- 11 Cuts and Negation 189**
  - 11.1 Green Cuts: Expressing Determinism 189
  - 11.2 Tail Recursion Optimization 195
  - 11.3 Negation 198
  - 11.4 Red Cuts: Omitting Explicit Conditions 202
  - 11.5 Default Rules 206
  - 11.6 Cuts for Efficiency 208
  - 11.7 Background 212
- 12 Extra-Logical Predicates 215**
  - 12.1 Input/Output 215
  - 12.2 Program Access and Manipulation 219
  - 12.3 Memo-Functions 221
  - 12.4 Interactive Programs 223
  - 12.5 Failure-Driven Loops 229
  - 12.6 Background 231
- 13 Program Development 233**
  - 13.1 Programming Style and Layout 233
  - 13.2 Reflections on Program Development 235
  - 13.3 Systematizing Program Construction 238
  - 13.4 Background 244

### ***III Advanced Prolog Programming Techniques 247***

- 14 Nondeterministic Programming 249**
  - 14.1 Generate-and-Test 249
  - 14.2 Don't-Care and Don't-Know Nondeterminism 263
  - 14.3 Artificial Intelligence Classics: ANALOGY, ELIZA, and McSAM 270
  - 14.4 Background 280
- 15 Incomplete Data Structures 283**
  - 15.1 Difference-Lists 283

15.2	Difference-Structures	291
15.3	Dictionaries	293
15.4	Queues	297
15.5	Background	300
<b>16</b>	<b>Second-Order Programming</b>	<b>301</b>
16.1	All-Solutions Predicates	301
16.2	Applications of Set Predicates	305
16.3	Other Second-Order Predicates	314
16.4	Background	317
<b>17</b>	<b>Interpreters</b>	<b>319</b>
17.1	Interpreters for Finite State Machines	319
17.2	Meta-Interpreters	323
17.3	Enhanced Meta-Interpreters for Debugging	331
17.4	An Explanation Shell for Rule-Based Systems	341
17.5	Background	354
<b>18</b>	<b>Program Transformation</b>	<b>357</b>
18.1	Unfold/Fold Transformations	357
18.2	Partial Reduction	360
18.3	Code Walking	366
18.4	Background	373
<b>19</b>	<b>Logic Grammars</b>	<b>375</b>
19.1	Definite Clause Grammars	375
19.2	A Grammar Interpreter	380
19.3	Application to Natural Language Understanding	382
19.4	Background	388
<b>20</b>	<b>Search Techniques</b>	<b>389</b>
20.1	Searching State-Space Graphs	389
20.2	Searching Game Trees	401
20.3	Background	407

**IV Applications 409**

<b>21</b>	<b>Game-Playing Programs</b>	<b>411</b>
21.1	Mastermind	411
21.2	Nim	415
21.3	Kalah	420
21.4	Background	423
<b>22</b>	<b>A Credit Evaluation Expert System</b>	<b>429</b>
22.1	Developing the System	429
22.2	Background	438
<b>23</b>	<b>An Equation Solver</b>	<b>439</b>
23.1	An Overview of Equation Solving	439
23.2	Factorization	448
23.3	Isolation	449
23.4	Polynomial	452
23.5	Homogenization	454
23.6	Background	457
<b>24</b>	<b>A Compiler</b>	<b>459</b>
24.1	Overview of the Compiler	459
24.2	The Parser	466
24.3	The Code Generator	470
24.4	The Assembler	475
24.5	Background	478
<b>A</b>	<b>Operators</b>	<b>479</b>
	<b>References</b>	<b>483</b>
	<b>Index</b>	<b>497</b>