
Contents

Preface	viii
1 Introduction	1
1.1 Introducing logic programming	2
2 Programming with relations	13
3 Recursive structures	21
3.1 Lists	21
3.2 Deriving facts about <i>append</i>	25
3.3 More relations on lists	29
3.4 Binary trees	32
4 The meaning of logic programs	35
4.1 Syntax	36
4.2 Truth tables	39
4.3 Adding functions and variables	41
4.4 Substitutions	42
5 Inference rules	47
5.1 Substitution and ground resolution	47
5.2 Refutation	50
5.3 Completeness	52
6 Unification and resolution	55
6.1 Unification	57
6.2 Resolution	62
6.3 Derivation trees and the lifting lemma	64
6.4 Completeness of resolution	66

7	SLD-resolution and answer substitutions	69
7.1	Linear resolution	70
7.2	SLD-resolution	73
7.3	Search trees	76
7.4	Answer substitutions	80
8	Negation as failure	85
8.1	Negation in goals	85
8.2	Negation in programs	87
8.3	Semantics of negation	88
9	Searching problems	91
9.1	Representing the problem	92
9.2	Avoiding cycles	94
9.3	Bounded and breadth-first search	96
10	Parsing	99
10.1	Arithmetic expressions	99
10.2	Difference lists	101
10.3	Expression trees	102
10.4	Grammar rules in Prolog	104
11	Evaluating and simplifying expressions	107
11.1	Evaluating expressions	107
11.2	Simplifying expressions	109
12	Hardware simulation	115
13	Program transformation	122
13.1	Unfolding and symbolic execution	122
13.2	Fold-unfold transformation	123
13.3	Improving the <i>reverse</i> program	125
14	About picoProlog	130
14.1	The picoProlog language	131
14.2	Built-in relations	132
14.3	The cut symbol	133
14.4	Implementation overview	136
15	Implementing depth-first search	139
15.1	Depth-first search	139
15.2	Representing the goal list	141
15.3	Representing goals	143
15.4	Answer substitutions	145

15.5	Depth-first search revisited	145
15.6	Choice points	149
15.7	Choosing representations	150
16	Representing terms and substitutions	153
16.1	Representing terms	153
16.2	Substitutions	155
16.3	Renaming	157
16.4	Printing terms	158
16.5	The trail	160
16.6	Unification	161
17	Implementation notes	164
17.1	Macros	165
17.2	String handling	168
17.3	Memory allocation	168
17.4	Symbol table	170
17.5	Lexical analysis	170
17.6	Syntax analysis	172
17.7	Trail	174
17.8	Unification	175
17.9	Interpreter	175
17.10	Built-in relations	175
17.11	Main program	176
18	Interpreter optimizations	178
18.1	Garbage collection	179
18.2	Indexing	180
18.3	Tail recursion	181
18.4	A concluding example	183
19	In conclusion	185
	Further reading	187
	Appendices	
A	Answers to the exercises	189
B	Using an ordinary Prolog system	203
C	PicoProlog source code	205
D	Cross-reference listing	244
	Index	249