CONTENTS

Preface v

1 Introduction 1

1.1 The origins of natural language processing 2
1.2 The imposition of structure 5
1.3 The representation of meaning 8
1.4 The role of knowledge 12
1.5 The emergence of a new technology 15
1.6 Using Prolog for natural language processing 17

2 Finite-state techniques 21

2.1 Finite-state transition networks 22
2.2 A notation for networks 27
2.3 Representing FSTNs in Prolog 37
2.4 Traversing FSTNs 40
2.5 Traversing FSTNs in Prolog 43
2.6 Finite-state transducers 50
2.7 Implementing FSTs in Prolog 57
2.8 Limitations of finite-state machines 59

3 Recursive and augmented transition networks 63

3.1 Recursive transition networks 64
3.2 Modelling recursion in English grammar 68
3.3 Representing RTNs in Prolog 72
3.4 Traversing RTNs 73
3.5 Implementing RTN traversal in Prolog 79
3.6 Pushdown transducers 82
3.7 Implementing pushdown transducers in Prolog 87
3.8 Advantages and limitations of RTNs 89
3.9 Augmented transition networks 91
3.10 Some reflections on ATNs 95

4	**Grammars**	**99**
4.1	Grammar as knowledge representation	100
4.2	Words, rules and structures	104
4.3	Representing simple grammars in Prolog	110
4.4	Subcategorization and the use of features	115
4.5	Definite clause grammars	127
4.6	Classes of grammars and languages	132
5	**Parsing, search and ambiguity**	**143**
5.1	A simple parsing problem	144
5.2	Bottom-up parsing	145
5.3	Top-down parsing	152
5.4	Parsing in Prolog	156
5.5	Comparing strategies	165
5.6	Breadth-first and depth-first search	166
5.7	Storing intermediate results	168
5.8	Ambiguity	169
5.9	Determinism and lookahead	174
6	**Well-formed substring tables and charts**	**179**
6.1	Well-formed substring tables	180
6.2	The active chart	189
6.3	The fundamental rule of chart parsing	193
6.4	Initialization	196
6.5	Rule invocation	196
6.6	Housekeeping	199
6.7	Implementing a simple bottom-up chart parser	200
6.8	Alternative rule invocation strategies	202
6.9	Implementing a simple top-down chart parser	204
6.10	Search strategy	205
6.11	Implementing flexible control	207
6.12	Efficiency	213
7	**Features and the lexicon**	**217**
7.1	Feature-theoretic syntax	218
7.2	Feature structures as graphs	221
7.3	Feature structures in Prolog	228
7.4	Subsumption and unification	230
7.5	The status of rules	238
7.6	Implementing PATR in Prolog	240
7.7	Chart parsing with feature-based grammars	248
7.8	Representation of lexical knowledge	256
7.9	Implementing a lexicon in Prolog	270
7.10	DAGs versus terms	273

8 Semantics 279

8.1 Compositionality 280
8.2 Meaning as reference 282
8.3 Translation to a meaning representation language 288
8.4 A database query language 290
8.5 Computational semantics as feature instantiation 293
8.6 Transitive verbs and quantification 294
8.7 Ambiguity, preferences and timing 301
8.8 Building semantic checking into the grammar 303

9 Question answering and inference 313

9.1 Question answering 314
9.2 Evaluating DBQ formulae 318
9.3 Standard logical inference 325
9.4 Implementing forwards inference in Prolog 333
9.5 The pathological nature of logical inference 339
9.6 Primitives and canonical forms 343
9.7 Classes and inheritance 347
9.8 Plausible inference and defaults 353

10 Pragmatics 359

10.1 Ambiguity and levels of language processing 360
10.2 Semantic and pragmatic roles of noun phrases 361
10.3 Given versus new information 364
10.4 Understanding by prediction 372
10.5 More controlled versions of prediction 376
10.6 Problems with prediction 381
10.7 Using discourse structure 382
10.8 Language generation as a goal-oriented process 386
10.9 Language understanding and plan recognition 398

Appendix Code listings 403

Solutions to selected exercises 469

Bibliography 473

Name Index 497

General Index 501