

Contents

1	An Overview of Language Processing	1
1.1	Linguistics and Language Processing	1
1.2	Applications of Language Processing	2
1.3	The Different Domains of Language Processing	4
1.4	Phonetics	5
1.5	Lexicon and Morphology	6
1.6	Syntax	8
1.6.1	Syntax as Defined by Noam Chomsky	8
1.6.2	Syntax as Relations and Dependencies	10
1.7	Semantics	11
1.8	Discourse and Dialogue	13
1.9	Why Speech and Language Processing Are Difficult	14
1.9.1	Ambiguity	15
1.9.2	Models and Their Implementation	16
1.10	An Example of Language Technology in Action: The Persona Project	17
1.10.1	Overview of Persona	17
1.10.2	The Persona's Modules	18
1.11	Further Reading	19
	Exercises	21
2	Corpus Processing Tools	23
2.1	Corpora	23
2.1.1	Types of Corpora	23
2.1.2	Corpora and Lexicon Building	25
2.1.3	Corpora as Knowledge Sources for the Linguist	27
2.2	Finite-State Automata	28
2.2.1	A Description	28
2.2.2	Mathematical Definition of Finite-State Automata	29
2.2.3	Finite-State Automata in Prolog	29
2.2.4	Deterministic and Nondeterministic Automata	31

2.2.5	Building a Deterministic Automaton from a Nondeterministic One	31
2.2.6	Searching a String with a Finite-State Automaton	32
2.2.7	Operations on Finite-State Automata	33
2.3	Regular Expressions	35
2.3.1	Repetition Metacharacters	37
2.3.2	The Dot Metacharacter	37
2.3.3	The Escape Character	37
2.3.4	The Longest Match	38
2.3.5	Character Classes	39
2.3.6	Nonprintable Symbols or Positions	41
2.3.7	Union and Boolean Operators	41
2.3.8	Operator Combination and Precedence	42
2.4	Programming with Regular Expressions	43
2.4.1	Perl	43
2.4.2	Strings and Regular Expressions in Perl	44
2.4.3	Matching	46
2.4.4	Substitutions	47
2.4.5	Translating Characters	47
2.4.6	String Operators	48
2.4.7	Back References	48
2.4.8	Predefined Variables	50
2.5	Finding Concordances	51
2.5.1	Concordances in Perl	51
2.5.2	Concordances in Prolog	55
2.6	Approximate String Matching	57
2.6.1	Edit Operations	57
2.6.2	Minimum Edit Distance	58
2.6.3	Computing the Minimum Edit Distance in Perl	59
2.6.4	Searching Edits in Prolog	60
2.7	Further Reading	62
	Exercises	64
3	Encoding and Annotation Schemes	65
3.1	Encoding Texts	65
3.2	Character Sets	66
3.2.1	Representing Characters	66
3.2.2	Unicode	68
3.2.3	Unicode Character Properties	69
3.2.4	The Unicode Encoding Schemes	73
3.3	Locales and Word Order	74
3.3.1	Presenting Time, Numerical Information, and Ordered Words	74
3.3.2	The Unicode Collation Algorithm	76

3.4	Markup Languages.....	77
3.4.1	A Brief Background	77
3.4.2	An Outline of XML.....	78
3.4.3	Writing a DTD	80
3.4.4	Writing an XML Document	83
3.4.5	Namespaces	84
3.4.6	XML and Databases	85
3.5	Further Reading	85
	Exercises	86
4	Topics in Information Theory and Machine Learning	87
4.1	Introduction	87
4.2	Codes and Information Theory	87
4.2.1	Entropy	87
4.2.2	Huffman Coding	89
4.2.3	Cross Entropy	93
4.2.4	Perplexity and Cross Perplexity	94
4.3	Entropy and Decision Trees	94
4.3.1	Machine Learning.....	94
4.3.2	Decision Trees	95
4.3.3	Inducing Decision Trees Automatically	96
4.4	Classification Using Linear Methods	98
4.4.1	Linear Classifiers	98
4.4.2	Choosing a Data Set	99
4.5	Linear Regression	99
4.5.1	Least Squares	100
4.5.2	The Gradient Descent.....	103
4.5.3	The Gradient Descent and Linear Regression	104
4.6	Linear Classification	107
4.6.1	An Example	107
4.6.2	Classification in an N -Dimensional Space	109
4.6.3	Linear Separability.....	110
4.6.4	Classification vs. Regression	110
4.7	Perceptron	111
4.7.1	The Heaviside Function	111
4.7.2	The Iteration.....	112
4.7.3	The Two-Dimensional Case	112
4.7.4	Stop Conditions	113
4.8	Support Vector Machines	113
4.8.1	Maximizing the Margin	113
4.8.2	Lagrange Multipliers	114
4.9	Logistic Regression	115
4.9.1	Fitting the Weight Vector	117
4.9.2	The Gradient Ascent	118

4.10	Encoding Symbolic Values as Numerical Features.....	119
4.11	Further Reading	120
	Exercises	121
5	Counting Words	123
5.1	Counting Words and Word Sequences	123
5.2	Text Segmentation	124
5.2.1	What Is a Word?	124
5.2.2	Breaking a Text into Words and Sentences	126
5.3	Tokenizing Words	126
5.3.1	Using White Spaces	127
5.3.2	Using White Spaces and Punctuation	127
5.3.3	Defining Contents.....	129
5.3.4	Tokenizing Texts in Prolog	129
5.3.5	Tokenizing Using Classifiers	130
5.4	Sentence Segmentation	132
5.4.1	The Ambiguity of the Period Sign	132
5.4.2	Rules to Disambiguate the Period Sign	132
5.4.3	Using Regular Expressions	132
5.4.4	Improving the Tokenizer Using Lexicons	133
5.4.5	Sentence Detection Using Classifiers	134
5.5	<i>N</i> -Grams	135
5.5.1	Some Definitions.....	135
5.5.2	A Crash Program to Count Words with Unix	135
5.5.3	Counting Unigrams in Prolog	137
5.5.4	Counting Unigrams with Perl	138
5.5.5	Counting Bigrams with Perl.....	139
5.6	Probabilistic Models of a Word Sequence	140
5.6.1	The Maximum Likelihood Estimation	140
5.6.2	Using ML Estimates with <i>Nineteen Eighty-Four</i>	142
5.7	Smoothing <i>N</i> -Gram Probabilities.....	144
5.7.1	Sparse Data.....	144
5.7.2	Laplace's Rule	145
5.7.3	Good-Turing Estimation	146
5.8	Using <i>N</i> -Grams of Variable Length	148
5.8.1	Linear Interpolation.....	149
5.8.2	Back-Off	150
5.8.3	Katz's Back-Off Model.....	151
5.9	Industrial <i>N</i> -Grams	152
5.10	Quality of a Language Model	153
5.10.1	Intuitive Presentation	153
5.10.2	Entropy Rate	153
5.10.3	Cross Entropy	154
5.10.4	Perplexity.....	155

5.11	Collocations	155
5.11.1	Word Preference Measurements	156
5.11.2	Extracting Collocations with Perl	159
5.12	Application: Retrieval and Ranking of Documents on the Web	161
5.12.1	Document Indexing	162
5.12.2	Representing Documents as Vectors	162
5.12.3	Vector Coordinates	163
5.12.4	Ranking Documents	164
5.12.5	Categorizing Text	165
5.13	Further Reading	166
	Exercises	166
6	Words, Parts of Speech, and Morphology	169
6.1	Words	169
6.1.1	Parts of Speech	169
6.1.2	Grammatical Features	170
6.1.3	Two Significant Parts of Speech: The Noun and the Verb	171
6.2	Lexicons	174
6.2.1	Encoding a Dictionary	175
6.2.2	Building a Trie in Prolog	176
6.2.3	Finding a Word in a Trie	179
6.3	Morphology	179
6.3.1	Morphemes	179
6.3.2	Morphs	181
6.3.3	Inflection and Derivation	181
6.3.4	Language Differences	185
6.4	Morphological Parsing	186
6.4.1	Two-Level Model of Morphology	186
6.4.2	Interpreting the Morphs	187
6.4.3	Finite-State Transducers	187
6.4.4	Conjugating a French Verb	189
6.4.5	Prolog Implementation	190
6.4.6	Application to Romance Languages	192
6.4.7	Ambiguity	192
6.4.8	Operations on Finite-State Transducers	193
6.5	Morphological Rules	194
6.5.1	Two-Level Rules	194
6.5.2	Rules and Finite-State Transducers	195
6.5.3	Rule Composition: An Example with French Irregular Verbs	197
6.6	The CoNLL Format	199

6.7	Application Examples	200
6.8	Further Reading	201
	Exercises	202
7	Part-of-Speech Tagging Using Rules	205
7.1	Resolving Part-of-Speech Ambiguity	205
7.1.1	A Manual Method	205
7.1.2	Which Method to Use to Automatically Assign Parts of Speech	206
7.2	Baseline	207
7.3	Tagging with Rules	208
7.3.1	Brill's Tagger	208
7.3.2	Implementation in Prolog	209
7.3.3	Deriving Rules Automatically	212
7.3.4	Confusion Matrices	213
7.4	Unknown Words	214
7.5	Standardized Part-of-Speech Tagsets	214
7.5.1	Multilingual Part-of-Speech Tags	215
7.5.2	Parts of Speech for English	217
7.5.3	An Annotation Scheme for Swedish	220
7.6	Further Reading	220
	Exercises	222
8	Part-of-Speech Tagging Using Statistical Techniques	223
8.1	Part-of-Speech Tagging with Linear Classifiers	223
8.2	The Noisy Channel Model	225
8.2.1	Presentation	225
8.2.2	The <i>N</i> -Gram Approximation	226
8.2.3	Tagging a Sentence	228
8.2.4	The Viterbi Algorithm: An Intuitive Presentation	229
8.3	Markov Models	230
8.3.1	Markov Chains	230
8.3.2	Trellis Representation	231
8.3.3	Hidden Markov Models	231
8.3.4	Three Fundamental Algorithms to Solve Problems with HMMs	233
8.3.5	The Forward Procedure	234
8.3.6	Viterbi Algorithm	236
8.3.7	The Backward Procedure	237
8.3.8	The Forward–Backward Algorithm	238
8.4	POS Tagging with the Perceptron	241
8.5	Tagging with Decision Trees	243
8.6	Unknown Words	244
8.7	An Application of the Noisy Channel Model: Spell Checking	245

8.8	A Second Application: Language Models for Machine Translation	246
8.8.1	Parallel Corpora	246
8.8.2	Alignment	246
8.8.3	Translation	249
8.8.4	Evaluating Translation	250
8.9	Further Reading	250
	Exercises	251
9	Phrase-Structure Grammars in Prolog	253
9.1	Using Prolog to Write Phrase-Structure Grammars	253
9.2	Representing Chomsky's Syntactic Formalism in Prolog	254
9.2.1	Constituents	254
9.2.2	Tree Structures	255
9.2.3	Phrase-Structure Rules	255
9.2.4	The Definite Clause Grammar (DCG) Notation	257
9.3	Parsing with DCGs	258
9.3.1	Translating DCGs into Prolog Clauses	258
9.3.2	Parsing and Generation	260
9.3.3	Left-Recursive Rules	261
9.4	Parsing Ambiguity	262
9.5	Using Variables	264
9.5.1	Gender and Number Agreement	264
9.5.2	Obtaining the Syntactic Structure	266
9.6	Application: Tokenizing Texts Using DCG Rules	268
9.6.1	Word Breaking	268
9.6.2	Recognition of Sentence Boundaries	269
9.7	Semantic Representation	270
9.7.1	λ -Calculus	270
9.7.2	Embedding λ -Expressions into DCG Rules	271
9.7.3	Semantic Composition of Verbs	273
9.8	An Application of Phrase-Structure Grammars and a Worked Example	274
9.9	Further Reading	278
	Exercises	278
10	Partial Parsing	281
10.1	Is Syntax Necessary?	281
10.2	Word Spotting and Template Matching	281
10.2.1	ELIZA	281
10.2.2	Word Spotting in Prolog	282
10.3	Named Entities and Multiwords	285
10.3.1	Named Entities	285
10.3.2	Multiwords	286
10.3.3	A Standard Named Entity Annotation	287

10.4	Detecting Named Entities with Rules	288
10.4.1	The Longest Match	289
10.4.2	Running the Program	290
10.5	Noun Groups and Verb Groups.....	291
10.5.1	Groups Versus Recursive Phrases	292
10.5.2	DCG Rules to Detect Noun Groups.....	293
10.5.3	DCG Rules to Detect Verb Groups.....	294
10.5.4	Running the Rules	295
10.6	Group Annotation Using Tags.....	298
10.6.1	Tagging Gaps	298
10.6.2	Tagging Words	299
10.6.3	Extending IOB to Two or More Groups	299
10.6.4	Annotation Examples from CoNLL 2000, 2002, and 2003	300
10.7	Machine Learning Methods to Detect Groups.....	301
10.7.1	Group Detection Using Symbolic Rules	301
10.7.2	Group Detection Using Stochastic Tagging	303
10.7.3	Using Classifiers	304
10.7.4	Group Detection Performance and Feature Engineering	306
10.8	Cascading Partial Parsers	307
10.9	Elementary Analysis of Grammatical Functions	307
10.9.1	Main Functions	307
10.9.2	Extracting Other Groups.....	308
10.10	An Annotation Scheme for Groups in French	311
10.11	Application: Information Extraction and the FASTUS System... ..	313
10.11.1	The Message Understanding Conferences	313
10.11.2	The Syntactic Layers of the FASTUS System.....	314
10.11.3	Evaluation of Information Extraction Systems	315
10.12	Further Reading	316
	Exercises	317
11	Syntactic Formalisms	321
11.1	Introduction	321
11.2	Chomsky's Grammar in Syntactic Structures	322
11.2.1	Constituency: A Formal Definition	323
11.2.2	Transformations.....	324
11.2.3	Transformations and Movements	326
11.2.4	Gap Threading	327
11.2.5	Gap Threading to Parse Relative Clauses.....	329
11.3	Standardized Phrase Categories for English	330
11.4	Unification-Based Grammars	332
11.4.1	Features.....	332
11.4.2	Representing Features in Prolog	333
11.4.3	A Formalism for Features and Rules.....	335

11.4.4	Features Organization	336
11.4.5	Features and Unification.....	338
11.4.6	A Unification Algorithm for Feature Structures.....	339
11.5	Dependency Grammars.....	341
11.5.1	Presentation	341
11.5.2	Properties of a Dependency Graph.....	344
11.6	Differences Between Tesnière’s Model and Current Conventions in Dependency Analysis	345
11.6.1	Prepositions and Auxiliaries.....	346
11.6.2	Coordination and Apposition.....	349
11.7	Valence	350
11.8	Dependencies and Functions	353
11.9	Corpus Annotation for Dependencies.....	355
11.9.1	Dependency Annotation Using XML.....	356
11.9.2	The CoNLL Annotation	357
11.10	Projectivization	359
11.10.1	A Prolog Program to Identify Nonprojective Arcs	360
11.10.2	A Method to Projectivize Links	363
11.11	From Constituency to Dependency	364
11.11.1	Transforming a Constituent Parse Tree into Dependencies	364
11.11.2	Trace Revisited.....	365
11.12	Further Reading	367
	Exercises	368
12	Constituent Parsing	371
12.1	Introduction	371
12.2	Bottom-Up Parsing	372
12.2.1	The Shift–Reduce Algorithm.....	372
12.2.2	Implementing Shift–Reduce Parsing in Prolog.....	373
12.2.3	Differences Between Bottom-Up and Top-Down Parsing	375
12.3	Chart Parsing.....	376
12.3.1	Backtracking and Efficiency	376
12.3.2	Structure of a Chart	376
12.3.3	The Active Chart.....	378
12.3.4	Modules of an Earley Parser	379
12.3.5	The Earley Algorithm in Prolog.....	382
12.3.6	The Earley Parser to Handle Left-Recursive Rules and Empty Symbols	386
12.4	Probabilistic Parsing of Context-Free Grammars	388
12.5	A Description of PCFGs.....	389
12.5.1	The Bottom-Up Chart	391
12.5.2	The Cocke–Younger–Kasami Algorithm in Prolog	393
12.5.3	Adding Probabilities to the CYK Parser	394

12.6	Evaluation of Constituent Parsers	395
12.6.1	Metrics	395
12.6.2	Performance of PCFG Parsing	396
12.7	Improving Probabilistic Context-Free Grammars	397
12.8	Lexicalized PCFG: Charniak’s Parser	398
12.9	Further Reading	400
	Exercises	401
13	Dependency Parsing	403
13.1	Introduction	403
13.2	Evaluation of Dependency Parsers	403
13.3	Nivre’s Parser	404
13.3.1	Extending the Shift–Reduce Algorithm to Parse Dependencies	404
13.3.2	Parsing an Annotated Corpus	405
13.3.3	Nivre’s Parser in Prolog	407
13.4	Guiding Nivre’s Parser	411
13.4.1	Parsing with Dependency Rules	411
13.4.2	Using Machine-Learning Techniques	415
13.5	Finding Dependencies Using Constraints	419
13.6	Covington’s Parser	420
13.6.1	Covington’s Nonprojective Parser	421
13.6.2	Relations Between Nivre’s and Covington’s Parsers ...	425
13.6.3	Covington’s Projective Parser	426
13.7	Eisner’s Parser	427
13.7.1	Adapting the CYK Parser to Dependencies	428
13.7.2	A More Efficient Version	431
13.7.3	Implementation	432
13.7.4	Learning Graphs with the Perceptron	433
13.8	Further Reading	435
	Exercises	436
14	Semantics and Predicate Logic	439
14.1	Introduction	439
14.2	Language Meaning and Logic: An Illustrative Example	440
14.3	Formal Semantics	441
14.4	First-Order Predicate Calculus to Represent the State of Affairs	442
14.4.1	Variables and Constants	442
14.4.2	Predicates	443
14.5	Querying the Universe of Discourse	444
14.6	Mapping Phrases onto Logical Formulas	445
14.6.1	Representing Nouns and Adjectives	446
14.6.2	Representing Noun Groups	446
14.6.3	Representing Verbs and Prepositions	447

14.7	The Case of Determiners	448
14.7.1	Determiners and Logic Quantifiers	448
14.7.2	Translating Sentences Using Quantifiers	448
14.7.3	A General Representation of Sentences	449
14.8	Compositionality to Translate Phrases to Logical Forms	451
14.8.1	Translating the Noun Phrase	452
14.8.2	Translating the Verb Phrase	453
14.9	Augmenting the Database and Answering Questions	454
14.9.1	Declarations	454
14.9.2	Questions with Existential and Universal Quantifiers.....	455
14.9.3	Prolog and Unknown Predicates	457
14.9.4	Other Determiners and Questions	457
14.10	Application: The Spoken Language Translator.....	458
14.10.1	Translating Spoken Sentences.....	458
14.10.2	Compositional Semantics	459
14.10.3	Semantic Representation Transfer	461
14.11	RDF and SPARQL as Alternatives to Prolog	462
14.11.1	RDF Triples	463
14.11.2	SPARQL.....	464
14.11.3	DBpedia and Yago	465
14.12	Further Reading	466
	Exercises	467
15	Lexical Semantics.....	469
15.1	Beyond Formal Semantics.....	469
15.1.1	La langue et la parole	469
15.1.2	Language and the Structure of the World	470
15.2	Lexical Structures	470
15.2.1	Some Basic Terms and Concepts	470
15.2.2	Ontological Organization.....	471
15.2.3	Lexical Classes and Relations	472
15.2.4	Semantic Networks	473
15.3	Building a Lexicon.....	474
15.3.1	The Lexicon and Word Senses	475
15.3.2	Verb Models.....	476
15.3.3	Definitions.....	477
15.4	An Example of Exhaustive Lexical Organization: WordNet.....	478
15.4.1	Nouns.....	479
15.4.2	Adjectives	480
15.4.3	Verbs.....	481
15.5	Automatic Word Sense Disambiguation	482
15.5.1	Senses as Tags	482
15.5.2	Associating a Word with a Context	483
15.5.3	Guessing the Topic.....	484

15.5.4	Naïve Bayes	484
15.5.5	Using Constraints on Verbs.....	485
15.5.6	Using Dictionary Definitions.....	486
15.5.7	An Unsupervised Algorithm to Tag Senses	487
15.5.8	Senses and Languages	488
15.6	Case Grammars	489
15.6.1	Cases in Latin	489
15.6.2	Cases and Thematic Roles.....	490
15.6.3	Parsing with Cases	492
15.6.4	Semantic Grammars	493
15.7	Extending Case Grammars	494
15.7.1	FrameNet.....	494
15.7.2	The Proposition Bank.....	495
15.7.3	Annotation of Syntactic and Semantic Dependencies	497
15.7.4	A Statistical Method to Identify Semantic Roles.....	500
15.8	An Example of Case Grammar Application: EVAR	505
15.8.1	EVAR's Ontology and Syntactic Classes	506
15.8.2	Cases in EVAR.....	506
15.9	Further Reading	506
	Exercises	508
16	Discourse	511
16.1	Introduction	511
16.2	Discourse: A Minimalist Definition.....	512
16.2.1	A Description of Discourse.....	512
16.2.2	Discourse Entities.....	512
16.3	References: An Application-Oriented View	513
16.3.1	References and Noun Phrases	513
16.3.2	Names and Named Entities.....	514
16.3.3	Finding Names – Proper Nouns.....	515
16.3.4	Disambiguation of Named Entities	516
16.4	Coreference	518
16.4.1	Anaphora.....	518
16.4.2	Solving Coreferences in an Example	519
16.4.3	The MUC Coreference Annotation	519
16.4.4	The CoNLL Coreference Annotation	521
16.5	References: A More Formal View	524
16.5.1	Generating Discourse Entities: The Existential Quantifier	524
16.5.2	Retrieving Discourse Entities: Definite Descriptions	524
16.5.3	Generating Discourse Entities: The Universal Quantifier	525

16.6	Solving Coreferences	527
16.6.1	A Simplistic Method: Using Syntactic and Semantic Compatibility	527
16.6.2	Solving Coreferences with Shallow Grammatical Information.....	528
16.6.3	Saliency in a Multimodal Context.....	529
16.6.4	Using a Machine-Learning Technique to Resolve Coreferences	531
16.6.5	More Complex Phenomena: Ellipses	535
16.7	Centering: A Theory on Discourse Structure	535
16.8	Discourse and Rhetoric	537
16.8.1	Ancient Rhetoric: An Outline	537
16.8.2	Rhetorical Structure Theory	538
16.8.3	Types of Relations	540
16.8.4	Implementing Rhetorical Structure Theory.....	540
16.9	Events and Time	543
16.9.1	Events	543
16.9.2	Event Types	544
16.9.3	Temporal Representation of Events	544
16.9.4	Events and Tenses.....	545
16.10	TimeML, an Annotation Scheme for Time and Events	548
16.11	Further Reading	549
	Exercises	551
17	Dialogue	553
17.1	Introduction	553
17.2	Why a Dialogue?.....	554
17.3	Architecture of a Dialogue System.....	554
17.4	Simple Dialogue Systems	555
17.4.1	Dialogue Systems Based on Automata	555
17.4.2	Dialogue Modeling	556
17.5	Speech Acts: A Theory of Language Interaction	558
17.6	Speech Acts and Human–Machine Dialogue.....	560
17.6.1	Speech Acts as a Tagging Model.....	560
17.6.2	Speech Acts Tags Used in the SUNDIAL Project.....	560
17.6.3	Dialogue Parsing	561
17.6.4	Interpreting Speech Acts	564
17.6.5	EVAR: A Dialogue Application Using Speech Acts	565
17.7	Taking Beliefs and Intentions into Account	567
17.7.1	Representing Mental States	568
17.7.2	The STRIPS Planning Algorithm	570
17.7.3	Causality	572
17.8	Further Reading	572
	Exercises	573

A	An Introduction to Prolog	575
A.1	A Short Background	575
A.2	Basic Features of Prolog	576
A.2.1	Facts	576
A.2.2	Terms	577
A.2.3	Queries	578
A.2.4	Logical Variables	579
A.2.5	Shared Variables	580
A.2.6	Data Types in Prolog	581
A.2.7	Rules	582
A.3	Running a Program	583
A.4	Unification	585
A.4.1	Substitution and Instances	585
A.4.2	Terms and Unification	585
A.4.3	The Herbrand Unification Algorithm	586
A.4.4	Example	587
A.4.5	The Occurs-Check	588
A.5	Resolution	589
A.5.1	Modus Ponens	589
A.5.2	A Resolution Algorithm	589
A.5.3	Derivation Trees and Backtracking	591
A.6	Tracing and Debugging	592
A.7	Cuts, Negation, and Related Predicates	594
A.7.1	Cuts	594
A.7.2	Negation	595
A.7.3	The <code>once/1</code> Predicate	597
A.8	Lists	597
A.9	Some List-Handling Predicates	598
A.9.1	The <code>member/2</code> Predicate	598
A.9.2	The <code>append/3</code> Predicate	599
A.9.3	The <code>delete/3</code> Predicate	600
A.9.4	The <code>intersection/3</code> Predicate	600
A.9.5	The <code>reverse/2</code> Predicate	601
A.9.6	The Mode of an Argument	602
A.10	Operators and Arithmetic	602
A.10.1	Operators	602
A.10.2	Arithmetic Operations	603
A.10.3	Comparison Operators	604
A.10.4	Lists and Arithmetic: The <code>length/2</code> Predicate	605
A.10.5	Lists and Comparison: The <code>quicksort/2</code> Predicate	606
A.11	Some Other Built-in Predicates	606
A.11.1	Type Predicates	606
A.11.2	Term Manipulation Predicates	608
A.12	Handling Run-Time Errors and Exceptions	609

A.13	Dynamically Accessing and Updating the Database	610
A.13.1	Accessing a Clause: The <code>clause/2</code> Predicate	610
A.13.2	Dynamic and Static Predicates	610
A.13.3	Adding a Clause: The <code>asserta/1</code> and <code>assertz/1</code> Predicates	611
A.13.4	Removing Clauses: The <code>retract/1</code> and <code>abolish/2</code> Predicates	612
A.13.5	Handling Unknown Predicates	612
A.14	All-Solutions Predicates	613
A.15	Fundamental Search Algorithms	614
A.15.1	Representing the Graph	615
A.15.2	Depth-First Search	616
A.15.3	Breadth-First Search	617
A.15.4	A* Search	618
A.16	Input/Output	618
A.16.1	Input/Output with Edinburgh Prolog	619
A.16.2	Input/Output with Standard Prolog	621
A.16.3	Writing Loops	623
A.17	Developing Prolog Programs	624
A.17.1	Presentation Style	624
A.17.2	Improving Programs	625
Exercises	627
References	631
Index	651