

CONTENTS

PREFACE xi

GLOSSARY OF SYMBOLS xv

1 FUNDAMENTALS 1

1.1 Basic Definitions 5

1.1.1 *Languages* 5

1.1.2 *Grammars* 7

1.1.3 *Classification of Grammars* 13

1.2 Ambiguity in Context-Free Grammars 14

1.2.1 *Bracketed Forms* 14

1.2.2 *Derivation Trees and Ambiguity* 17

1.2.3 *Examples of Ambiguous Grammars* 19

1.2.4 *A Negative Result* 24

Exercises 25

1.3 Graph Searching 27

1.3.1 *Developing an Algorithm* 29

1.3.2 *Choice of Data Structures and Implementation* 32

1.3.3 *Checking the Solution* 35

1.3.4 *Summing Up* 39

1.4 Useless Productions 40

1.4.1 *Rephrasing the Definition* 41

1.4.2 *Accessible Non-terminals* 42

1.4.3 *Terminating Productions* 43

Exercises 57

Bibliographic Notes 57

2 REGULAR LANGUAGES 59

- 2.1 Transition Diagrams and Non-Deterministic Machines 60
- 2.2 Regular Expressions 63
- 2.3 Converting a Non-Deterministic Machine into a Deterministic Machine 71
 - 2.3.1 *The Language Recognized by a Deterministic Machine* 73
 - 2.3.2 *The Language Recognized by a Non-deterministic Machine* 75
 - 2.3.3 *The Conversion Algorithm—An Example* 76
 - 2.3.4 *A Rigorous Development of the Conversion Algorithm* 78
- 2.4 Numbers in ALGOL 60 81
- Appendix—Proof of Theorem 2.1 86
- Exercises 87
- Bibliographic Notes 90

3 LL PARSING 92

- 3.1 Syntax Analysis by Recursive Descent 92
- 3.2 Checking the Strong $LL(k)$ Condition 102
 - 3.2.1 *Calculating $FIRST_k$ Sets* 104
 - 3.2.2 *Calculating $FOLLOW_k$ Sets* 107
 - 3.2.3 *A Complete Example* 110
 - 3.2.4 *A Test for Ambiguity* 115
- 3.3 The Special Case $k=1$ 116
 - 3.3.1 *Graph-Searching and the $FIRST$ and $FOLLOW$ sets* 117
 - 3.3.2 *An Efficient Algorithm* 123
 - 3.3.3 *Implementing the Algorithm* 125
- 3.4 $LL(k)$ Grammars 138
- 3.5 Practical Considerations 147
 - 3.5.1 *Eliminating Terminal Procedures* 147
 - 3.5.2 *Extend BNF* 148
 - 3.5.3 *Overcoming Non- $LL(1)$ Features* 151
- Exercises 152
- Bibliographic Notes 155

4 LR PARSING 157

- 4.1 $LR(0)$ Grammars 158
 - 4.1.1 *The Basic Parsing Algorithm* 158
 - 4.1.2 *Calculating the $LR(0)$ Contexts* 161
 - 4.1.3 *Developing the Parsing Algorithm* 164
 - 4.1.4 *The $LR(0)$ Condition* 168

4.2	LR(k) and SLR(k) Grammars	172
4.2.1	LR(k) Grammars	172
4.2.2	Simple LR(k) Grammars	176
4.3	Theoretical Comparisons	181
	Exercises	186
	Bibliographic Notes	187

5 ERROR REPAIR 189

5.1	String-to-String Repair	190
5.1.1	Edit Operations	190
5.1.2	The "Best" Repair	191
5.1.3	A Formal Model	195
5.1.4	Equational Characterization	197
5.2	Repair of Regular Languages	199
5.2.1	An Example	199
5.2.2	The Formal Model	202
5.2.3	Equational Characterization	203
5.3	Repair of Context-Free Languages	204
5.4	Algorithms for Finding Least-Cost Repairs	210
5.4.1	Dijkstra's Algorithm	212
5.4.2	Gaussian Elimination	214
	Exercises	216
	Bibliographic Notes	217

6 ERROR RECOVERY 219

6.1	A Simple Recovery Scheme	221
6.2	Our Objective	227
6.3	Locally Optimal Repairs	233
6.3.1	The Definition	234
6.3.2	Production Positions	237
6.3.3	Repairs and Actions	239
6.3.4	Program 6.2 Again	241
6.3.5	Passing Parameters	242
6.3.6	A Summary	247
6.4	Does It Work?	247
6.5	Error Recovery in a PL/0 Parser	254
	Exercises	270
	Bibliographic Notes	271

7 LIMITATIONS 273

- 7.1 The *uvwxy* Theorem 273
- 7.2 Semantic Ambiguities in ALGOL 60 279
 - 7.2.1 *Side Effects* 279
 - 7.2.2 *for Statements* 280
- 7.3 Syntactic Ambiguities in PASCAL 282
 - 7.3.1 *Types* 283
 - 7.3.2 *Scope Rules* 284
- 7.4 Conclusion 287
- Exercises 287
- Bibliographic Notes 288

INDEX 293