

Contents

1	Introduction	1
1.1	Intended Scope and Audience	1
1.2	Compiler Parts and Corresponding Concepts	2
2	Syntax	5
2.1	Introduction	5
2.1.1	Artificial and Formal Languages	5
2.1.2	Language Types	6
2.1.3	Chapter Outline	7
2.2	Formal Language Theory	8
2.2.1	Alphabet and Language	8
2.2.2	Language Operations	11
2.2.3	Set Operations	13
2.2.4	Star and Cross	14
2.2.5	Quotient	17
2.3	Regular Expressions and Languages	17
2.3.1	Definition of Regular Expression	18
2.3.2	Derivation and Language	20
2.3.3	Other Operators	23
2.3.4	Closure Properties of <i>REG</i> Family	24
2.4	Linguistic Abstraction	25
2.4.1	Abstract and Concrete Lists	26
2.5	Context-Free Generative Grammars	30
2.5.1	Limits of Regular Languages	30
2.5.2	Introduction to Context-Free Grammars	31
2.5.3	Conventional Grammar Representations	33
2.5.4	Derivation and Language Generation	35
2.5.5	Erroneous Grammars and Useless Rules	37
2.5.6	Recursion and Language Infinity	39
2.5.7	Syntax Trees and Canonical Derivations	40
2.5.8	Parenthesis Languages	44

2.5.9	Regular Composition of Context-Free Languages	46
2.5.10	Ambiguity	47
2.5.11	Catalogue of Ambiguous Forms and Remedies	49
2.5.12	Weak and Structural Equivalence	57
2.5.13	Grammar Transformations and Normal Forms	59
2.6	Grammars of Regular Languages	67
2.6.1	From Regular Expressions to Context-Free Grammars	67
2.6.2	Linear Grammars	69
2.6.3	Linear Language Equations	71
2.7	Comparison of Regular and Context-Free Languages	73
2.7.1	Limits of Context-Free Languages	76
2.7.2	Closure Properties of <i>REG</i> and <i>CF</i>	78
2.7.3	Alphabetic Transformations	80
2.7.4	Grammars with Regular Expressions	83
2.8	More General Grammars and Language Families	87
2.8.1	Chomsky Classification	87
3	Finite Automata as Regular Language Recognizers	93
3.1	Introduction	93
3.2	Recognition Algorithms and Automata	94
3.2.1	A General Automaton	95
3.3	Introduction to Finite Automata	98
3.4	Deterministic Finite Automata	100
3.4.1	Error State and Total Automata	101
3.4.2	Clean Automata	101
3.4.3	Minimal Automata	103
3.4.4	From Automata to Grammars	106
3.5	Nondeterministic Automata	108
3.5.1	Motivation of Nondeterminism	108
3.5.2	Nondeterministic Recognizers	110
3.5.3	Automata with Spontaneous Moves	112
3.5.4	Correspondence between Automata and Grammars	114
3.5.5	Ambiguity of Automata	115
3.5.6	Left-Linear Grammars and Automata	116
3.6	Directly from Automata to Regular Expressions: BMC Method	117
3.7	Elimination of Nondeterminism	119
3.7.1	Construction of Accessible Subsets	121
3.8	From Regular Expression to Recognizer	124
3.8.1	Thompson Structural Method	124
3.8.2	Algorithm of Glushkov, McNaughton and Yamada	126
3.8.3	Deterministic Recognizer by Berry and Sethi Algorithm	134
3.9	Regular Expressions with Complement and Intersection	137
3.9.1	Product of Automata	138
3.10	Summary of Relations between Regular Languages, Grammars, and Automata	143

4	Pushdown Automata and Top-down Parsing	147
4.1	Introduction	147
4.1.1	Pushdown Automaton	148
4.1.2	From Grammar to Pushdown Automaton	149
4.1.3	Definition of Pushdown Automaton	152
4.2	One Family for Context-Free Languages and Pushdown Automata	157
4.2.1	Intersection of Regular and Context-Free Languages	160
4.2.2	Deterministic Pushdown Automata and Languages	160
4.3	Syntax Analysis	170
4.3.1	Top-Down and Bottom-Up Analysis	170
4.3.2	Grammar as Network of Finite Automata	172
4.3.3	Nondeterministic Recognition Algorithm	176
4.4	Top-Down Deterministic Syntax Analysis	178
4.4.1	Condition for $LL(1)$ Parsing	179
4.4.2	How to Obtain $LL(1)$ Grammars	192
4.4.3	Increasing Look-ahead	196
5	Bottom-Up and General Parsing	203
5.1	Introduction	203
5.2	Bottom-Up Deterministic Syntax Analysis	203
5.2.1	$LR(0)$ Method	204
5.2.2	$LR(0)$ Grammars	206
5.2.3	Shift-Reduce Parser	211
5.2.4	Syntax Analysis with $LR(k)$ Look-Ahead	214
5.2.5	$LR(1)$ Parsing Algorithm	226
5.2.6	Properties of $LR(k)$ Language and Grammar Families	226
5.2.7	How to Obtain $LR(1)$ Grammars	229
5.2.8	$LR(1)$ Parsing with Extended Context-Free Grammars	232
5.2.9	Comparison of Deterministic Families REG , $LL(k)$, and $LR(k)$	241
5.3	A General Parsing Algorithm	242
5.3.1	Introductory Example	243
5.3.2	Earley Algorithm	247
5.3.3	Computational Complexity	252
5.3.4	Handling of Empty Rules	254
5.3.5	Further Developments	256
5.4	How to Choose a Parser	256
6	Translation Semantics and Static Analysis	259
6.1	Introduction	259
6.1.1	Chapter Outline	260
6.2	Translation Relation and Function	262
6.3	Transliteration	264
6.4	Regular Translations	264

6.4.1	Two-Input Automaton	266
6.4.2	Translation Functions and Finite Transducers	270
6.5	Purely Syntactic Translation	275
6.5.1	Infix and Polish Notations	277
6.5.2	Ambiguity of Source Grammar and Translation	281
6.5.3	Translation Grammars and Pushdown Transducers	282
6.5.4	Syntax Analysis with Online Translation	287
6.5.5	Top-Down Deterministic Translation	287
6.5.6	Bottom-Up Deterministic Translation	290
6.5.7	Comparisons	295
6.5.8	Closure Properties of Translations	296
6.6	Semantic Translations	297
6.6.1	Attribute Grammars	299
6.6.2	Left and Right Attributes	301
6.6.3	Definition of Attribute Grammar	305
6.6.4	Dependence Graph and Attribute Evaluation	307
6.6.5	One Sweep Semantic Evaluation	311
6.6.6	Other Evaluation Methods	315
6.6.7	Combined Syntax and Semantic Analysis	316
6.6.8	Typical Applications of Attribute Grammars	324
6.7	Static Program Analysis	334
6.7.1	A Program as an Automaton	334
6.7.2	Liveness Intervals of Variables	338
6.7.3	Reaching Definitions	345
	References	353
	Index	357



<http://www.springer.com/978-1-84882-049-4>

Formal Languages and Compilation

Crespi Reghizzi, S.

2009, XII, 368 p. 100 illus., Hardcover

ISBN: 978-1-84882-049-4