

Contents

Chapter 1 Introduction	1
1.1 Language and Mankind	1
1.2 Language and Computer	3
1.3 Compilation of Programming Languages	12
1.4 Number of Passes of Compiler	17
1.5 An Example of Compilation of a Statement	19
1.6 Organization of the Book	21
Problems	23
References	23
Chapter 2 Grammars and Languages	25
2.1 Motivation of the Chapter	25
2.2 Preliminary Knowledge	25
2.3 Grammar	27
2.4 Language	31
2.5 Language Generated by a Grammar	34
2.6 Turing Machine	37
2.7 Issues Concerning Grammars and Languages	52
Problems	53
References	54
Chapter 3 Finite State Automata and Regular Languages	55
3.1 Motivations of the Chapter	55
3.2 Languages, Grammars and Automata	55
3.3 Deterministic Finite Automata	59
3.4 Nondeterministic Finite Automata	64
3.5 Regular Expressions	65
3.6 Regular Grammar	66
3.7 Kleene's and Moore's Theorems	68
3.8 Pumping Theorems and Closure Properties for L_{REG}	69

3.9 Applications of Finite Automata	70
3.10 Variants of Finite Automata	72
Problems	77
References	78
Chapter 4 Lexical Analysis	79
4.1 Motivation of the Chapter	79
4.2 Lexical Analyzer	80
4.2.1 Role of Lexical Analyzer	81
4.2.2 Identifier Analysis	84
4.2.3 Handling of Constants	86
4.2.4 Structure of Lexical Analyzer	87
4.3 Output of Lexical Analyzer	95
4.4 Error Handling	97
Problems	98
References	98
Chapter 5 Push-Down Automata and Context-Free Languages	101
5.1 Motivation of the Chapter	101
5.2 Push-Down Automata	102
5.3 Context-Free Languages (L_{CF})	103
5.4 Pumping Theorems for Context-Free Languages	105
5.5 Push-Down Automata and Context-Free Languages	106
5.6 Applications of Context-Free Languages	106
5.7 Turing Machines	107
5.8 Turing Machines as Language Accepters	108
5.9 Equivalence of Various Turing Machines	115
5.10 Recursively Enumerable Languages (L_{RE})	116
5.11 Context-Sensitive Languages (L_{CS})	117
5.12 Hierarchy of Machines, Grammars and Languages	119
5.12.1 Hierarchy of Machines	119
5.12.2 Hierarchy of Grammars and Languages	120
5.13 Relations Among Machines, Languages and Grammars	121
Problems	124
References	124
Chapter 6 Context-Free Grammars	125
6.1 Motivation of the Chapter	125
6.2 Context-Free Grammars	126
6.3 Characteristics of Context-Free Grammars	135
Problems	154

References	155
Chapter 7 Syntax Analysis	157
7.1 Motivation of the Chapter	157
7.2 Role of Syntax Analysis in Compilers	158
7.3 Methods of Syntax Analysis	161
7.4 LL(1) Syntactical Analysis Method	173
7.5 Bottom-Up Syntactical Analysis Method	180
7.6 LR(1) Syntactical Analysis Method	185
7.6.1 LR(0) Syntactical Analysis	185
7.6.2 SLR(1) Syntactical Analysis	189
7.6.3 LALR(1) Syntactical Analysis	191
7.6.4 LR(1) Syntactical Analysis	193
7.6.5 Comparison Between LL(1) Syntactical Analysis Method and LR(1) Syntactical Analysis Method	202
Problems	205
References	206
Chapter 8 Attribute Grammars and Analysis	207
8.1 Motivation of the Chapter	207
8.2 Attribute Grammar	208
8.3 Dependence Graph and Evaluation of Attributes	212
8.3.1 Dynamic Attribute Evaluation	217
8.3.2 Loop Handling	221
8.4 L Attribute Grammas and S Attribute Grammars	222
Problems	225
References	227
Chapter 9 Algebraic Method of Compiler Design	229
9.1 Motivation of the Chapter	229
9.2 Source Language	230
9.3 Algebraic Foundation and Reasoning Language	238
9.3.1 Algebra Fundamentals	239
9.3.2 Reasoning Language	247
9.4 A Simple Compiler	275
9.4.1 The Normal Form	276
9.4.2 Normal Form Reduction	277
9.4.3 The Target Machine	281
Problems	282
References	282

Chapter 10	Generation of Intermediate Code	285
10.1	Motivation of the Chapter	285
10.2	Intermediate Code Languages	286
10.2.1	Graphic Representation	287
10.2.2	Postfix Representation	290
10.2.3	The Quadruple Code	292
	Problems	311
	References	312
Chapter 11	Debugging and Optimization	313
11.1	Motivation of the Chapter	313
11.2	Errors Detection and Recovery	313
11.3	Debugging of Syntax Errors	316
11.3.1	Error Handling of LL(1) Parser	318
11.3.2	Error Handling in LR(1) Analysis	319
11.4	Semantic Error Check	319
11.5	Optimization of Programs	320
11.6	Principal Ways of Optimization	324
11.6.1	Elimination of Subexpressions	324
11.6.2	Copy Propagation	325
11.6.3	Dead-Code Elimination	326
11.6.4	Loop Optimization	327
11.6.5	Reduction of Strength	328
	Problems	329
	References	330
Chapter 12	Storage Management	331
12.1	Motivation of the Chapter	331
12.2	Global Allocation Strategy	332
12.3	Algorithms for Allocation	334
12.3.1	Algorithm for Stack Allocation	334
12.3.2	Algorithm for Heap Allocation	336
12.4	Reclamation of Used Space	337
12.4.1	Basic Garbage Collection Algorithm	338
12.4.2	Supports to Garbage Collector From Compilers	340
12.4.3	Reference Counts	342
12.4.4	Tokens and Scans	343
12.4.5	Dual Space Copy	344
12.4.6	Contract	345
12.5	Parameter Passing	346
12.5.1	Call-by-Value	347

12.5.2	Call-by-References	347
12.5.3	Copy-Restore	348
12.5.4	Call-by-Name	348
Problems		349
References		351
Chapter 13	Generation of Object Code	353
13.1	Motivation of the Chapter	353
13.2	Issues of Design of Generators of Target Codes	354
13.2.1	Input of Code Generators	354
13.2.2	Target Programs	355
13.2.3	Storages Management	355
13.2.4	Selection of Instructions	356
13.2.5	Register Allocation	357
13.2.6	Selection of Order of Computation	358
13.2.7	Method of Generation of Codes	358
13.3	Target Machine MMIX	358
13.3.1	Binary Bits and Bytes	359
13.3.2	Memory and Registers	361
13.3.3	Instructions	362
13.3.4	Load and Store	363
13.3.5	Arithmetic Operations	365
13.3.6	Conditional Instructions	367
13.3.7	Bit Operations	368
13.3.8	Byte Operations	369
13.3.9	Jumps and Branches	373
13.3.10	Subprogram Calls	375
13.3.11	Interruptions	377
13.4	Assembly Language of MMIX	382
13.5	Generation of MMIXAL Target Codes	389
13.5.1	Translation of Expressions in Reversed Polish Form	390
13.5.2	Translation of Triple Expressions	390
13.5.3	Translation of Expression Quadruples	391
13.5.4	Translation of Expressions	392
13.5.5	Translation of Syntax Tree Form of Expressions	393
13.5.6	Translation of Various Statements	394
Problems		395
References		397

Chapter 14	Compilation of Object-oriented Languages	399
14.1	Motivation of the Chapter	399
14.2	Objects and Compilation	400
14.3	Characteristics of Objects	403
14.3.1	Inheritance	403
14.3.2	Method Overload	404
14.3.3	Polymorphic	405
14.3.4	Dynamic Constraint	406
14.3.5	Multiple Inheritances	408
14.3.6	Multiple Inheritances of Inter-reliance	410
	Problems	412
	References	413
Chapter 15	Compilation of Parallel Languages	415
15.1	Motivation of the Chapter	415
15.2	Rising of Parallel Computers and Parallel Computation	415
15.3	Parallel Programming	419
15.3.1	Shared Variables and Monitors	420
15.3.2	Message Passing Model	422
15.4	Object-oriented Languages	424
15.5	Linda Meta Array Space	425
15.6	Data Parallel Languages	427
15.7	Code Generation for Hidden Parallel Programs	428
15.7.1	Types of Regions	430
15.7.2	Formation of Regions	431
15.7.3	Schedule Algorithms for Regions	436
	Problems	437
	References	437
Chapter 16	Compilation of Grid Computing	439
16.1	Motivation of the Chapter	439
16.2	Rising of Grid Computing and Intent	439
16.3	Grid Computing Model	442
16.3.1	Group Routing	443
16.3.2	Routing in Linear Array	445
16.4	Compilation of Grid Computing	447
	Problems	450
	References	450
Index		451