

Table of Contents

Preface	ix
Chapter 1 Overview of Compilers and Language Translation.....	1
1.1 The Role of Programming Languages	1
1.2 Translators and Compilers	1
1.3 Tombstone Diagrams	4
1.4 Bootstrapping a Compiler	8
1.5 Interpreters.....	10
1.6 The Compiler Project.....	11
1.7 Essential Terms and Concepts.....	14
1.8 Exercises	14
Chapter 2 Structure of a Compiler.....	17
2.1 Scanner.....	18
2.2 Parser	18
2.3 Constraint Analyzer	19
2.4 Code Generator	19
2.5 Optimizer.....	20
2.6 Final Code Generator.....	21
2.7 Tables and Maps.....	21
2.8 Error Handler	22
2.9 Passes	23
2.10 Compiler Design Goals	23
2.11 Essential Terms and Concepts.....	24
2.12 Exercises	25
Chapter 3 Context-Free Grammars.....	27
3.1 Specifying a Programming Language.....	27
3.2 Context-Free Grammars	28
3.3 Alternate Rule Notations	35
3.4 Grammar Transformations	38
3.5 Derivations and Parse Trees	39
3.6 Abstract Syntax Trees	43
3.7 A Context-Free Grammar for Context-Free Grammars.....	44
3.8 Essential Terms and Concepts.....	45
3.9 Exercises	46

Chapter 4 The Programming Language CPRL	49
4.1 General Lexical Considerations.....	49
4.2 Declarations, Statements, and Expressions	50
4.3 Types	51
4.4 Statements.....	54
4.5 Programs	56
4.6 Subprograms	57
4.7 Essential Terms and Concepts.....	59
4.8 Exercises	60
Chapter 5 Lexical Analysis (a.k.a. Scanning)	63
5.1 Class Position	63
5.2 Class Source	63
5.3 Class Symbol	65
5.4 Class Token	66
5.5 Class CharUtil	67
5.6 Class ErrorHandler.....	67
5.7 Class BoundedBuffer.....	68
5.8 Class Scanner	69
5.9 Handling Lexical Errors	76
5.10 Testing Class Scanner.....	78
5.11 Essential Terms and Concepts.....	80
5.12 Exercises	80
Chapter 6 Syntax Analysis (a.k.a. Parsing).....	83
6.1 Example: Implementing Method <code>parseLoopStmt()</code>	84
6.2 Recursive Descent Parsing	87
6.3 First and Follow Sets.....	93
6.4 LL(k) Grammars and Recursive Descent Parsing.....	100
6.5 Variables versus Variable Expressions	102
6.6 Handling Grammar Limitations	102
6.7 Scope and Visibility	105
6.8 Class IdTable	108
6.9 Parsing Variables and Variable Expressions	114
6.10 Class Parser	115
6.11 Essential Terms and Concepts.....	116
6.12 Exercises	117

Chapter 7 Error Handling/Recovery	119
7.1 Types of Compilation Errors	119
7.2 Handling Errors	121
7.3 Error Recovery	122
7.4 Additional Error Recovery Strategies.....	128
7.5 Essential Terms and Concepts.....	129
7.6 Exercises	129
Chapter 8 Abstract Syntax Trees.....	131
8.1 Overview of Abstract Syntax Trees	131
8.2 Structure of Abstract Syntax Trees.....	135
8.3 Extending Scopes with References to Declarations	137
8.4 Types and Declarations.....	143
8.5 Structural References versus Nonstructural References	145
8.6 Determining Types of Variables and Expressions	148
8.7 Maintaining Context During Parsing	150
8.8 Essential Terms and Concepts.....	152
8.9 Exercises	153
Chapter 9 Constraint Analysis.....	155
9.1 Overview of Constraint Analysis.....	155
9.2 Constraint Rules for CPRL/ \emptyset	159
9.3 Examples of Constraint Analysis	161
9.4 Essential Terms and Concepts.....	165
9.5 Exercises	165
Chapter 10 The CPRL Virtual Machine	167
10.1 Overview of the CVM	167
10.2 Brief Comparison with the JVM	169
10.3 CVM Uses Relative Addressing	170
10.4 Loading a Program into Memory	170
10.5 Using the Stack to Hold Temporary Values	173
10.6 Essential Terms and Concepts.....	179
10.7 Exercises	180
Chapter 11 Code Generation.....	181
11.1 Overview of Code Generation	181
11.2 Labels and Branching.....	182
11.3 Load and Store Instructions.....	186
11.4 Computing Relative Addresses for Variables	188

11.5 Expressions	189
11.6 Statements	193
11.7 Initial Declarations	197
11.8 Disassembler	198
11.9 Essential Terms and Concepts	200
11.10 Exercises	200
Chapter 12 Code Optimization	203
12.1 Overview of Code Optimization	203
12.2 Common Optimizations	206
12.3 Optimization in CPRL	211
12.4 Essential Terms and Concepts	213
12.5 Exercises	213
Chapter 13 Subprograms	215
13.1 Review of Subprograms, Scope, and Parameters	215
13.2 Run-time Organization for Subprograms	219
13.3 Activation Record	221
13.4 Parameters	228
13.5 Subprogram Calls and Returns	230
13.6 Calling Conventions for CPRL on the CVM	231
13.7 Computing Relative Addresses	232
13.8 Example of Program Execution	236
13.9 Essential Terms and Concepts	248
13.10 Exercises	248
Chapter 14 Arrays	251
14.1 Using CPRL Arrays	251
14.2 Implementing CPRL Arrays	253
14.3 Essential Terms and Concepts	259
14.4 Exercises	259
Chapter 15 Strings	261
15.1 Using CPRL Strings	261
15.2 Implementing CPRL Strings	263
15.3 Essential Terms and Concepts	272
15.4 Exercises	272

Chapter 16 Records	275
16.1 Using CPRL Records	275
16.2 Implementing CPRL Records	279
16.3 Essential Terms and Concepts.....	283
16.4 Exercises	284
Appendix A The Compiler Project	287
Appendix B Additional Project Exercises	297
Appendix C Definition of the Programming Language CPRL.....	301
C.1 Lexical Considerations.....	301
C.2 Types	303
C.3 Constants and Variables.....	305
C.4 Operators and Expressions	307
C.5 Statements	308
C.6 Programs.....	311
C.7 Subprograms.....	311
Appendix D The CPRL Grammar	315
Appendix E Definition of the CPRL Virtual Machine	319
E.1 Specification.	319
E.2 Implementation	320
E.3 CVM Instruction Set Architecture	323
Appendix F Searching for Reserved Words	333
F.1 Benchmarking the Search Algorithms	334
F.2 Sequential Search 1.....	335
F.3 Sequential Search 2.....	336
F.4 Sequential Search 3.....	337
F.5 Binary Search	338
F.6 Search by Length.....	339
F.7 Search by First Character	341
F.8 Gperf Hash Search	342
F.9 Search Using Switch Expression	345
F.10 Search Using HashMap.....	346
Appendix G JIT Compilation versus AOT Compilation.....	347
Annotated Compiler References and Websites.....	351
Index.....	355

