<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2</td>
<td>AN ATTRIBUTE GRAMMAR FOR WREN</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>The Symbol Table</td>
<td>74</td>
</tr>
<tr>
<td></td>
<td>Commands</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>Expressions</td>
<td>82</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>90</td>
</tr>
<tr>
<td>3.3</td>
<td>LABORATORY: CONTEXT CHECKING WREN</td>
<td>92</td>
</tr>
<tr>
<td></td>
<td>Declarations</td>
<td>96</td>
</tr>
<tr>
<td></td>
<td>Commands</td>
<td>99</td>
</tr>
<tr>
<td></td>
<td>Expressions</td>
<td>101</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
<td>102</td>
</tr>
<tr>
<td>3.4</td>
<td>FURTHER READING</td>
<td>103</td>
</tr>
</tbody>
</table>

Chapter 4

TWO-LEVEL GRAMMARS | 105 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>CONCEPTS AND EXAMPLES</td>
</tr>
<tr>
<td></td>
<td>Fortran String Literals</td>
</tr>
<tr>
<td></td>
<td>Derivation Trees</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
<tr>
<td>4.2</td>
<td>A TWO-LEVEL GRAMMAR FOR WREN</td>
</tr>
<tr>
<td></td>
<td>Declarations</td>
</tr>
<tr>
<td></td>
<td>Commands and Expressions</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
<tr>
<td>4.3</td>
<td>TWO-LEVEL GRAMMARS AND PROLOG</td>
</tr>
<tr>
<td></td>
<td>Implementing Two-Level Grammars in Prolog</td>
</tr>
<tr>
<td></td>
<td>Two-Level Grammars and Logic Programming</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
<tr>
<td>4.4</td>
<td>FURTHER READING</td>
</tr>
</tbody>
</table>

Chapter 5

THE LAMBDA CALCULUS | 139 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>CONCEPTS AND EXAMPLES</td>
</tr>
<tr>
<td></td>
<td>Syntax of the Lambda Calculus</td>
</tr>
<tr>
<td></td>
<td>Curried Functions</td>
</tr>
<tr>
<td></td>
<td>Semantics of Lambda Expressions</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
<tr>
<td>5.2</td>
<td>LAMBDA REDUCTION</td>
</tr>
<tr>
<td></td>
<td>Reduction Strategies</td>
</tr>
<tr>
<td></td>
<td>Correlation with Parameter Passing</td>
</tr>
<tr>
<td></td>
<td>Constants in the Pure Lambda Calculus</td>
</tr>
<tr>
<td></td>
<td>Functional Programming Languages</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
<tr>
<td>5.3</td>
<td>LABORATORY: A LAMBDA CALCULUS EVALUATOR</td>
</tr>
<tr>
<td></td>
<td>Scanner and Parser</td>
</tr>
<tr>
<td></td>
<td>The Lambda Calculus Evaluator</td>
</tr>
<tr>
<td></td>
<td>Exercises</td>
</tr>
</tbody>
</table>
5.4 FURTHER READING 166

Chapter 6
SELF-DEFINITION OF PROGRAMMING LANGUAGES 167

6.1 SELF-DEFINITION OF LISP 167
Metacircular Interpreter 169
Running the Interpreter 174
Exercises 178

6.2 SELF-DEFINITION OF PROLOG 179
Displaying Failure 181
Exercises 185

6.3 FURTHER READING 185

Chapter 7
TRANSLATIONAL SEMANTICS 187

7.1 CONCEPTS AND EXAMPLES 187
A Program Translation 189
Exercises 191

7.2 ATTRIBUTE GRAMMAR CODE GENERATION 191
Expressions 193
Commands 201
Exercises 213

7.3 LABORATORY: IMPLEMENTING CODE GENERATION 215
Commands 217
Expressions 219
Exercises 221

7.4 FURTHER READING 222

Chapter 8
TRADITIONAL OPERATIONAL SEMANTICS 223

8.1 CONCEPTS AND EXAMPLES 224
VDL 226
Exercises 227

8.2 SECD: AN ABSTRACT MACHINE 228
Example 231
Parameter Passing 232
Static Scoping 233
Exercises 234

8.3 LABORATORY: IMPLEMENTING THE SECD MACHINE 235
Exercises 237

8.4 STRUCTURAL OPERATIONAL SEMANTICS: INTRODUCTION 238
Specifying Syntax 239
Inference Systems and Structural Induction 242
Exercises 244

8.5 STRUCTURAL OPERATIONAL SEMANTICS: EXPRESSIONS 245
Semantics of Expressions in Wren 245
CONTENTS

Example 248
Outcomes 250
Exercises 252

8.6 STRUCTURAL OPERATIONAL SEMANTICS: COMMANDS 253
A Sample Computation 256
Semantic Equivalence 260
Natural Semantics 261
Exercises 262

8.7 LABORATORY: IMPLEMENTING STRUCTURAL OPERATIONAL SEMANTICS 264
Commands 265
Expressions 267
Top-Level Driver 268
Exercises 269

8.8 FURTHER READING 269

Chapter 9
DENOTATIONAL SEMANTICS 271

9.1 CONCEPTS AND EXAMPLES 271
The Syntactic World 272
The Semantic World 273
Compositionality 276
Exercises 277

9.2 A CALCULATOR LANGUAGE 277
Calculator Semantics 280
Semantic Functions 282
A Sample Calculation 283
Exercises 284

9.3 THE DENOTATIONAL SEMANTICS OF WREN 285
Semantic Domains 286
Language Constructs in Wren 288
Auxiliary Functions 290
Semantic Equations 290
Error Handling 293
Semantic Equivalence 294
Input and Output 294
Elaborating a Denotational Definition 296
Exercises 302

9.4 LABORATORY: IMPLEMENTING DENOTATIONAL SEMANTICS 304
Exercises 309

9.5 DENOTATIONAL SEMANTICS WITH ENVIRONMENTS 310
Environments 311
Stores 312
Semantic Functions 313
Semantic Equations 316
Procedures 318
Exercises 321
CONTENTS xv

9.6 CHECKING CONTEXT-SENSITIVE SYNTAX 323
Exercises 327

9.7 CONTINUATION SEMANTICS 328
Continuations 331
The Programming Language Gull 333
Auxiliary Functions 335
Semantic Equations 336
The Error Continuation 336
Exercises 338

9.8 FURTHER READING 339

Chapter 10
DOMAIN THEORY AND FIXED-POINT SEMANTICS 341

10.1 CONCEPTS AND EXAMPLES 341
Recursive Definitions of Functions 342
Recursive Definitions of Sets (Types) 343
Modeling Nontermination 344
Exercises 345

10.2 DOMAIN THEORY 345
Elementary Domains 348
Product Domains 349
Sum Domains (Disjoint Unions) 351
Function Domains 355
Continuity of Functions on Domains 361
Exercises 363

10.3 FIXED-POINT SEMANTICS 365
First Step 366
Second Step 368
Continuous Functionals 374
Fixed points for Nonrecursive Functions 379
Revisiting Denotational Semantics 380
Fixed-Point Induction 382
Exercises 384

10.4 LABORATORY: RECURSION IN THE LAMBDA CALCULUS 388
Conditional Expressions 390
Paradoxical Combinator 390
Fixed-Point Identity 392
Exercises 393

10.5 FURTHER READING 394

Chapter 11
AXIOMATIC SEMANTICS 395

11.1 CONCEPTS AND EXAMPLES 395
Axiomatic Semantics of Programming Languages 396

11.2 AXIOMATIC SEMANTICS FOR WREN 398
Assignment Command 398
Input and Output 400
CONTENTS

Rules of Inference 401
While Command and Loop Invariants 405
More on Loop Invariants 408
Nested While Loops 410
Exercises 415

11.3 AXIOMATIC SEMANTICS FOR PELICAN 418
Blocks 420
Nonrecursive Procedures 422
Recursive Procedures 425
Exercises 429

11.4 PROVING TERMINATION 432
Steps in Showing Termination 433
Termination of Recursive Procedures 435
Exercises 436

11.5 INTRODUCTION TO PROGRAM DERIVATION 437
Table of Cubes 437
Binary Search 440
Exercises 441

11.6 FURTHER READING 442

Chapter 12

ALGEBRAIC SEMANTICS 443

12.1 CONCEPTS AND EXAMPLES 444
A Module for Truth Values 446
Module Syntax 447
A Module for Natural Numbers 448
A Module for Characters 452
A Parameterized Module and Some Instantiations 453
A Module for Finite Mappings 456
Exercises 459

12.2 MATHEMATICAL FOUNDATIONS 460
Ground Terms 461
Σ-Algebras 461
A Congruence from the Equations 463
The Quotient Algebra 465
Homomorphisms 466
Consistency and Completeness 467
Exercises 469

12.3 USING ALGEBRAIC SPECIFICATIONS 471
Data Abstraction 471
A Module for Unbounded Queues 472
Implementing Queues as Unbounded Arrays 474
Verification of Queue Axioms 477
ADTs As Algebras 477
Abstract Syntax and Algebraic Specifications 481
Exercise 485
12.4 ALGEBRAIC SEMANTICS FOR WREN 487
  Types and Values in Wren 488
  Abstract Syntax for Wren 489
  A Type Checker for Wren 490
  An Interpreter for Wren 494
  A Wren System 498
  Exercises 499

12.5 LABORATORY: IMPLEMENTING ALGEBRAIC SEMANTICS 499
  Module Booleans 500
  Module Naturals 501
  Declarations 503
  Commands 503
  Expressions 505
  Exercises 505

12.6 FURTHER READING 506

Chapter 13
ACTION SEMANTICS 507

13.1 CONCEPTS AND EXAMPLES 508
  Data and Sorts 511
  Yielders 514
  Actions 515
    The Functional Facet 515
    The Imperative Facet 518
  Exercises 520

13.2 ACTION SEMANTICS OF A CALCULATOR 522
  Semantic Functions 523
  Semantic Equations 524
  A Sample Calculation 528
  Exercises 530

13.3 THE DECLARATIVE FACET AND WREN 531
  The Programming Language Wren 534
  Exercises 540

13.4 THE REFLECTIVE FACET AND PELICAN 541
  The Reflective Facet and Procedures 545
  Procedures Without Parameters 547
  Procedures With A Parameter 548
  Recursive Definitions 550
  Translating to Action Notation 551
  Exercises 558

13.5 LABORATORY: TRANSLATING INTO ACTION NOTATION 559
  Exercises 563

13.6 FURTHER READING 563
## Contents

### Appendix A

**Logic Programming with Prolog** 565

- Prolog 566
- BNF Syntax for Prolog 568
- A Prolog Example 569
- Predefined Predicates 571
- Recursion in Prolog 572
- Control Aspects of Prolog 574
- Lists in Prolog 575
- Sorting in Prolog 581
- The Logical Variable 582
- Equality and Comparison in Prolog 583
- Input and Output Predicates 585

### Appendix B

**Functional Programming with Scheme** 587

- Lisp 588
- Scheme Syntax 589
- Functions on S-expressions 590
- Lists in Scheme 591
- Syntax for Functions 592
- Scheme Evaluation 593
- Special Forms 596
- Defining Functions in Scheme 596
- Recursive Definitions 598
- Lambda Notation 599
- Recursive Functions on Lists 599
- Scope Rules in Scheme 603
- Proving Correctness in Scheme 605
- Higher-Order Functions 606
- Currying 608
- Tail Recursion 609

### Bibliography

611

### Index

625