

Contents

Part I Specification Fundamentals

1	The Role of Specification	3
1.1	Software Complexity	4
1.1.1	Size Complexity	6
1.1.2	Structural Complexity	6
1.1.3	Environmental Complexity	7
1.1.4	Application Domain Complexity	9
1.1.5	Communication Complexity	10
1.2	Software Specification	11
1.2.1	What is a Specification?	12
1.2.2	Why Specify?	12
1.2.3	What to Specify?	13
1.2.4	When to Specify?	14
1.2.5	How to Control Complexity?	15
1.2.6	A Critique of Natural Language Specification	18
1.3	Exercises	18
1.4	Bibliographic Notes	19
	References	21
2	Specification Activities	23
2.1	Integrating Formal Methods into the Software Life-Cycle	24
2.2	Administrative and Technical Roles	28
2.2.1	Specification Roles	28
2.2.2	Design Roles	29
2.2.3	Implementation Roles	30
2.3	Exercises	31
2.4	Bibliographic Notes	32
	References	32
3	Specification Qualities	35
3.1	Process Quality	36

3.1.1	Why a Programming Language Cannot Serve as a Specification Language?	36
3.1.2	Attributes of Formal Specification Languages	38
3.1.3	A Model of Process Quality	40
3.2	Product Quality and Utility	40
3.2.1	Conformance to Stated Goals	41
3.2.2	Quality Dimensions and Quality Model	43
3.3	Exercises	44
3.4	Bibliographic Notes	45
	References	45
4	Abstraction	47
4.1	What Is Abstraction?	47
4.2	Abstractions in Mathematics	48
4.3	Fundamental Abstractions in Computing	48
4.4	Abstractions for Software Construction	50
4.4.1	Problem Abstractions	51
4.4.2	Domain Abstraction	51
4.4.3	Environmental Abstraction	53
4.4.4	System Abstractions	54
4.5	Exercises	55
4.6	Bibliographic Notes	56
	References	56
 Part II Formalism Fundamentals		
5	Formal Systems	61
5.1	Peano’s Axiomatization of Naturals—Formalization in Mathematics	62
5.2	Model and Theory	63
5.2.1	Formalization in Engineering	63
5.2.2	Formalization in Science	63
5.2.3	Formalization Process in Software Engineering	64
5.3	Components of a Formal System	65
5.3.1	Syntax	65
5.3.2	Semantics	66
5.3.3	Inference Mechanism	67
5.4	Properties of Formal Systems	69
5.4.1	Consistency	70
5.4.2	Completeness	70
5.4.3	Decidability	71
5.5	Extended Syntactic Metalanguage	71
5.6	Exercises	74
5.7	Bibliographic Notes	76
	References	76

6 Automata	77
6.1 Deterministic Finite Accepters	78
6.1.1 State Machine Modeling	79
6.2 Nondeterministic Finite Accepters	85
6.2.1 Finite State Transducers	93
6.3 Exercises	101
6.4 Bibliographic Notes	102
References	103
7 Extended Finite State Machine	105
7.1 State Machine Hierarchy	107
7.1.1 Menu-Driven User Interface Model	110
7.2 Modularity and Bottom-up Construction	113
7.2.1 Simulation	118
7.3 Transition Points	119
7.4 Case Study—Elevator Control	120
7.5 Exercises	124
7.6 Bibliographic Notes	127
References	127
8 Classification of Formal Specification Methods	129
8.1 The Four Pillars	129
8.2 Classification	130
8.2.1 Property-Oriented Specification Methods	130
8.2.2 Model-Based Specification Techniques	131
8.3 Languages Chosen for Discussion	132
8.4 Bibliographic Notes	133
References	133
Part III Logic	
9 Propositional Logic	137
9.1 Syntax and Semantics	137
9.2 Proof	139
9.2.1 Reasoning Based on Adopting a Premise	139
9.2.2 Inference Based on Natural Deduction	140
9.2.3 Proof by Resolution	141
9.3 Consistency and Completeness	143
9.4 Exercises	144
9.5 Bibliographic Notes	145
References	145
10 Predicate Logic	147
10.1 Syntax and Semantics	148
10.1.1 Semantics	149
10.2 Validity, Equality, and Equivalence	151

10.2.1	Equality and Equivalence	151
10.3	More on Quantified Expressions	154
10.3.1	Policy Language Specification	155
10.3.2	Knowledge Representation	158
10.4	Proofs	160
10.4.1	Natural Deduction Process	160
10.4.2	Resolution	162
10.4.3	Decidability	165
10.5	Axiomatic Specification Examples	166
10.5.1	Hoare's Notation	166
10.6	Exercises	171
10.7	Bibliographic Notes	174
	References	174
11	Temporal Logic	177
11.1	Temporal Logic for Specification and Verification	178
11.2	Concept of World and Notion of Time	179
11.2.1	Temporal Abstraction	179
11.2.2	Discrete or Continuous	180
11.2.3	Linear and Branching Models of Time	181
11.2.4	Further Specializations of Time	181
11.3	Propositional Temporal Logic (PTL)	181
11.3.1	Syntax	182
11.3.2	Model and Semantics	183
11.3.3	Formal Semantics	184
11.3.4	More Temporal Operators	184
11.3.5	Axioms	186
11.3.6	Formalizing Properties in PTL	187
11.3.7	Specifications	189
11.4	First Order Temporal Logic (FOTL)	195
11.4.1	Formalizing Properties in FOTL	196
11.4.2	Temporal Logic Semantics of Sequential Programs	199
11.4.3	Temporal Logic Semantics of Concurrent Systems with Shared Variables	201
11.5	Formal Verification	205
11.5.1	Verification of Simple FOTL Specifications	205
11.5.2	Model Checking	210
11.5.3	Program Graphs, Transition Systems, and Kripke Structures	212
11.5.4	Model Checking using Büchi Automata	217
11.6	Exercises	221
11.7	Bibliographic Notes	226
	References	228

Part IV Mathematical Abstractions for Model-Based Specifications

12 Set Theory and Relations	233
12.1 Formal Specification Based on Set Theory	233
12.1.1 Set Notation	234
12.1.2 Reasoning with Sets	235
12.1.3 A Specification Example	237
12.2 Formal Specification Based on Relations and Functions	242
12.2.1 Relations and Functions	242
12.2.2 Functions on Relations	244
12.2.3 Reasoning	248
12.2.4 A Specification Example	251
12.3 Formal Specification Based on Sequences	254
12.3.1 Notation	254
12.3.2 Sequence Operators	254
12.3.3 Proofs	257
12.3.4 A Specification Example	261
12.4 Exercises	262
12.5 Bibliographic Notes	263
References	264

Part V Property-Oriented Specifications

13 Algebraic Specification	267
13.1 Algebra and Specification	267
13.2 Algebras—A Brief Introduction	270
13.2.1 Homomorphisms	271
13.3 Abstract Data Types	273
13.3.1 Presentation	274
13.3.2 Semantics	276
13.4 Properties of Algebraic Specifications	277
13.4.1 Reasoning	277
13.4.2 Extending Many-Sorted Specifications	279
13.4.3 Classification of Operations	280
13.4.4 Adequacy	281
13.5 Structured Specifications	282
13.6 OBJ3—An Algebraic Specification Language	286
13.6.1 OBJ3 Basic Syntax	288
13.6.2 Built-In Sorts and Subsorts	290
13.7 Signature and Equations	294
13.7.1 Signature of a Module	295
13.7.2 Equations	296
13.8 Parameterized Programming	296
13.8.1 Theories	297
13.8.2 Views	298
13.8.3 Parameterized Modules	298

13.8.4	Instantiation	299
13.8.5	Module Expression	301
13.9	Case Study—A Multiple Window Environment	302
13.9.1	Requirements	302
13.9.2	Modeling	303
13.9.3	Formal Specifications	303
13.10	Exercises	309
13.11	Bibliographic Notes	310
	References	311
14	Larch	313
14.1	The Two Tiers of Larch	313
14.2	LSL—Larch Shared Language	315
14.2.1	Equational Specification	315
14.2.2	More Expressive Specifications and Stronger Theories	319
14.2.3	Composing Traits	321
14.2.4	Renaming	321
14.2.5	Stating Checkable Properties	322
14.2.6	Stating Assumptions	324
14.2.7	Operator Overloading	326
14.2.8	In-line Traits	327
14.3	More LSL Examples	329
14.3.1	File	330
14.3.2	Date and Zone	333
14.3.3	Time	336
14.4	Larch/C++: A Larch Interface Specification Language for C++	339
14.4.1	Relating Larch/C++ to C++	341
14.4.2	Function Specification	346
14.4.3	Additional Function Specification Features	348
14.5	Proofs in LSL	348
14.5.1	Proof Obligations	349
14.5.2	LP, the Larch Prover	351
14.6	Case Study—Two Examples from Rogue Wave Library	355
14.6.1	RWZone Specification	355
14.6.2	RWFile Specification	356
14.7	Exercises	358
14.8	Bibliographic Notes	363
	References	363
15	Calculus of Communicating Systems	365
15.1	Why a Specific Calculus for Concurrency Is Necessary?	367
15.2	Informal Introduction to CCS	368
15.3	CCS—Syntax and Semantics	377
15.3.1	Syntax	377
15.3.2	The Operational Semantics of Agents	378

15.4	Simulation and Equivalence	383
15.4.1	Derivation Trees	384
15.4.2	Milner's Laws	387
15.4.3	Labeled Transition Systems—Some Properties	391
15.4.4	Trace Equivalence	392
15.4.5	Equivalence and Congruence	394
15.5	Exercises	399
15.6	Bibliographic Notes	401
	References	402

Part VI Model-Based Specifications

16	Vienna Development Method	405
16.1	Structure of a VDM Specification	405
16.2	Representational Abstraction	406
16.2.1	Identifiers	407
16.2.2	Simple Types	407
16.2.3	Composite Types	409
16.2.4	Patterns, Bindings and Values	416
16.2.5	State Representation	417
16.2.6	Invariants	420
16.3	Operational Abstraction	421
16.3.1	Let Expression	421
16.3.2	Function Definitions	422
16.3.3	Operation Definitions	424
16.4	Statements	427
16.5	Specification Examples	430
16.6	Case Study—Computer Network	440
16.7	Rigorous Reasoning	447
16.8	Refinement and Proof Obligations	449
16.8.1	Data Refinement	449
16.8.2	Example for Data Refinement	451
16.8.3	Operation Decomposition	453
16.8.4	Example for Operation Decomposition	454
16.9	Exercises	455
16.10	Bibliographic Notes	457
	References	458
17	The Z Notation	461
17.1	Abstractions in Z	461
17.2	Representational Abstraction	461
17.2.1	Types	462
17.2.2	Abbreviation	464
17.2.3	Relations and Functions	465
17.2.4	Sequences	466

17.2.5	Bags	467
17.2.6	Free Types	470
17.2.7	Schemas	471
17.2.8	State Representation	481
17.3	Operational Abstraction	482
17.3.1	Operations	482
17.3.2	Schema Decorators and Conventions	484
17.3.3	Sequential Composition	487
17.3.4	Functions	488
17.3.5	Generic Functions	489
17.4	Specification Examples	490
17.5	Proving Properties from Z Specifications	505
17.5.1	Initial State Validation	506
17.5.2	Consistency of Operations	509
17.6	Case Study: An Automated Billing System	513
17.7	Additional Features in Z	521
17.7.1	Precondition Calculation	522
17.7.2	Promotion	524
17.8	Refinement and Proof Obligations	526
17.8.1	Data Refinement	527
17.8.2	Proof Obligations	531
17.9	Exercises	534
17.10	Bibliographic Notes	536
	References	537
18	The Object-Z Specification Language	539
18.1	Basic Structure of an Object-Z Specification	539
18.1.1	Parameterized Class	542
18.2	Distinguished Features of Object-Orientation	544
18.2.1	Encapsulation	544
18.2.2	Inheritance	544
18.2.3	Polymorphism	547
18.3	Composition of Operations	548
18.3.1	Sequential Composition Operator	548
18.3.2	Concurrency Operator	549
18.3.3	Parallel Communication Operator	550
18.3.4	Nondeterministic Choice Operator	551
18.3.5	Environment Enrichment Operator	551
18.4	Specification Examples	552
18.5	Case Study	564
18.6	Exercises	571
18.7	Bibliographic Notes	573
	References	574

19 The B-Method	577
19.1 Abstract Machine Notation (AMN)	577
19.1.1 Structure of a B Specification	578
19.2 Notations	582
19.2.1 Arrays	582
19.3 Nondeterministic Statements	584
19.3.1 ANY Statement	584
19.3.2 CHOICE Statement	585
19.3.3 SELECT Statement	586
19.3.4 PRE Statement	586
19.4 Structured Specifications	587
19.4.1 The INCLUDES Clause	587
19.4.2 The USES Clause	591
19.4.3 The SEES Clause	594
19.5 Refinement	596
19.5.1 Sequential Composition of Statements	596
19.5.2 Local Variables	597
19.5.3 Refinement Machine	597
19.6 Specification Examples	600
19.7 Case Study—A Ticketing System in a Parking Lot	613
19.8 Proof Obligations	623
19.8.1 Proof Obligations for INCLUDES Clause	626
19.8.2 Proof Obligations for USES Clause	627
19.8.3 Proof Obligations for SEES Clause	628
19.8.4 Proof Obligations for Refinement	628
19.9 Exercises	630
19.10 Bibliographic Notes	631
References	632
Index	635



<http://www.springer.com/978-0-85729-276-6>

Specification of Software Systems

Alagar, V.S.; Periyasamy, K.

2011, XXVI, 646 p., Hardcover

ISBN: 978-0-85729-276-6