

Contents

Part I A Foundation Course in CSP

1	Building a Simple Sequential Process	3
1.1	Basic Operators	3
1.1.1	Prefixing	3
1.1.2	Recursion	4
1.1.3	Guarded Alternative	5
1.2	Choice Operators	10
1.2.1	External Choice	10
1.2.2	Nondeterministic Choice	11
1.2.3	Conditional Choice	12
1.2.4	Multi-Part Events: Extending the Notation of Channels	14
1.3	A Few Important Processes	16
1.4	Traces Refinement	16
1.5	What is a Communication?	18
1.6	Tools	19
1.6.1	Finite-State Machines	21
2	Understanding CSP	23
2.1	Algebra	23
2.2	The Traces Model and Traces Refinement	29
2.2.1	Working out $traces(P)$	30
2.2.2	Traces and Laws	33
2.2.3	Unique Fixed Points	34
2.2.4	Specification and Refinement	36
2.2.5	After and Initials	40
2.3	Operational Semantics and Labelled Transition Systems	41
2.4	Tools	43
3	Parallel Operators	45
3.1	Synchronous Parallel	45
3.1.1	Turning Parallel Processes into Sequential Ones	46

3.2	Alphabetised Parallel	49
3.3	Interleaving	57
3.4	Generalised Parallel	59
3.5	Parallel Composition as Conjunction	61
3.6	Tools	64
3.7	Postscript: On Alphabets	65
4	CSP Case Studies	67
4.1	Sudoku in CSP	67
4.2	Deadlock-Free Routing	73
4.2.1	Head for the Trees!	75
4.2.2	Uncloggable Rings	80
4.2.3	The Mad Postman	84
4.3	Communications Protocols	86
5	Hiding and Renaming	93
5.1	Hiding	93
5.1.1	The Consequences of Hiding	97
5.1.2	Hiding <i>versus</i> Constructiveness	100
5.2	Renaming and Alphabet Transformations	102
5.2.1	Injective Functions	103
5.2.2	Non-injective Functions	104
5.2.3	Relational Renaming	105
5.3	Linking Operators	109
5.4	Tools	112
6	Beyond Traces	115
6.1	A Brief Introduction to Failures and Divergences	115
6.2	Failures and Divergences in Specifications	120
6.3	Ungranted Requests and the Limits of Failures	123
6.4	Avoiding Divergence	124
6.5	Abstraction by Hiding	125
6.6	Tools	129
7	Further Operators	131
7.1	Termination and Sequential Composition	131
7.1.1	What is Termination?	131
7.1.2	Distributed Termination	136
7.1.3	Effects on the Failures-Divergences Model	137
7.2	Interrupting Processes	138
7.3	Tools	141
8	Using FDR	143
8.1	What is FDR?	143
8.1.1	Running and Debugging Checks	146
8.1.2	FDR's Settings	148

8.1.3	Defining Non-Process Objects	150
8.1.4	The Limits of FDR	155
8.2	Checking Parallel Processes	157
8.3	The Structure of a Refinement Check	161
8.4	Failures and Divergences	163
8.5	Watchdogs	166
8.6	Breadth versus Depth, Symmetry and Divergence	166
8.7	Determinism Checking	170
8.8	Compression	173
8.8.1	Using Compression	175
8.9	Notes and Reflections	184

Part II Theory

9	Operational Semantics	191
9.1	Transition Systems and State Machines	192
9.2	Firing Rules for CSP	200
9.2.1	SOS Style Operational Rules	201
9.2.2	Combinator Style Operational Rules	204
9.3	From Combinators to Supercombinators!	212
9.4	Translating Combinators to CSP	214
9.5	Relationships with Abstract Models	221
9.5.1	Extracting Failures and Divergences	221
9.5.2	Infinite Traces and Infinite Branching	222
9.6	Tools	227
9.7	Notes	227
10	Denotational Semantics and Behavioural Models	229
10.1	Introduction	229
10.1.1	Fixed-Point Theory	231
10.2	Analysing Traces Semantics	232
10.3	The Stable Failures Model	236
10.3.1	Applications	239
10.3.2	Channel-Based Failures	241
10.4	The Failures-Divergences Model	241
10.5	Determinism, Confluence and Proof	247
10.6	Full Abstraction and Congruence	251
10.7	Notes	252
11	Finite Observation Models	255
11.1	What is a Behavioural Model?	255
11.1.1	The Finest Model of Them All	256
11.1.2	Clouding the Glass	258
11.2	A Tour through the Hierarchy	259
11.2.1	The Ready Sets, or Acceptances Model \mathcal{A}	260
11.2.2	The Stable Refusal Testing Model \mathcal{RT}	262

11.2.3	The Stable Revivals Model	264
11.2.4	Other Models	266
11.3	The Big Picture	268
11.4	Tools	270
12	Infinite-Behaviour Models	271
12.1	Divergence-Strict Models for Finite Nondeterminism	272
12.1.1	Determinism amongst the Richer Models	274
12.1.2	Applications	275
12.2	Strict Divergence for General CSP	276
12.2.1	Healthiness Conditions	276
12.2.2	Fixed Point Theories for Unbounded Nondeterminism	278
12.2.3	Applying Infinite-Behaviour Models	280
12.3	The Hierarchy of Divergence-Strict Models	282
12.4	Seeing Beyond Divergence	283
12.4.1	Applications of \mathcal{M}^\sharp	285
12.5	The Fixed Point Theory of \mathcal{M}^\sharp	286
12.6	The Hierarchy	289
12.7	Tools	290
12.8	Notes	290
13	The Algebra of CSP	293
13.1	Introduction	293
13.2	AOS Form	295
13.3	Algebraic Operational Semantics	299
13.4	Normal Forms	305
13.5	A Tour through the Hierarchy	309
13.6	General Programs	315
13.7	Notes	317
Part III Using CSP in Practice		
14	Timed Systems 1: <i>tock</i>-CSP	321
14.1	Introduction	321
14.2	A Brief History of Time in CSP	321
14.3	<i>tock</i> -CSP	323
14.3.1	Expressing Timing Constraints	324
14.4	Case Study: Bully Algorithm	327
14.4.1	Part 1: Specification	329
14.4.2	Part 2: Implementation	330
14.4.3	Analysis	335
14.4.4	Conclusions	337
14.5	Maximal Progress and Priority	339
14.5.1	Case Study: Timed Routing	340
14.6	Specifying and Metering <i>tock</i> -CSP	341
14.7	Tools	343

15	Timed Systems 2: Discrete Timed CSP	345
15.1	Modelling Timed CSP	345
15.1.1	Semantics	349
15.2	Examples	351
15.3	Digitisation	353
15.4	Notes for Chapters 14 and 15	355
16	More About FDR	357
16.1	Normalisation	357
16.2	More About Compression	363
16.2.1	Strong Bisimulation	364
16.2.2	DRW-Bisimulation	364
16.2.3	τ -Loop Elimination	365
16.2.4	Diamond Elimination	367
16.2.5	Laziness and chase	370
16.3	Handling Large Checks	373
16.3.1	Memory Locality	373
16.3.2	Parallel Implementation	376
16.4	Generalising the Specification Model	379
17	State Explosion and Parameterised Verification	385
17.1	Induction	386
17.1.1	The Limitations of Induction	390
17.2	Data Independence	390
17.2.1	Thresholds	392
17.2.2	Beyond Thresholds	397
17.3	Data-Independent Induction	398
17.4	Buffer Tolerance	407
17.4.1	Definitions, Basics and Tree Networks	408
17.4.2	Functional and Confluent Processes	411
17.5	Approximation Based Methods	413
17.6	Notes	415

Part IV Exploring Concurrency

18	Shared-Variable Programs	419
18.1	Writing a Compiler in CSP_M	421
18.1.1	Data Types	422
18.1.2	Variable Names	423
18.1.3	Compilation Strategy	424
18.1.4	Compiling a Thread	425
18.1.5	Evaluating an Expression	427
18.2	Applying Compression	430
18.3	The Front End of SVA	434
18.3.1	SVL	434
18.3.2	The GUI	436

18.3.3 Possible Extensions	436
18.4 Specifications in SVA	438
18.5 Case Study: Lamport’s Bakery Algorithm	439
18.6 Case Study: The Dining Philosophers in SVL	441
18.7 Notes	446
19 Understanding Shared-Variable Concurrency	447
19.1 Dirty Variables	447
19.2 Case Study: Simpson’s 4-Slot Algorithm	451
19.3 Observing and Refining SVL Programs	457
19.3.1 The Bakery Algorithm Revisited	464
19.4 Atomic Equivalent Programs	465
19.5 Overseers: Modelling Complex Data-Types	469
19.6 Abstraction in SVA	471
19.6.1 Two Threads out of Many	471
19.6.2 Finitely Representing an Infinite Linear Order	474
19.7 Notes	480
20 Priority and Mobility	481
20.1 Case Study: Knight’s Tour	481
20.2 Priority	486
20.2.1 Statecharts	490
20.2.2 Synchronous Two-Phase Automata	494
20.3 Mobility	497
20.3.1 An Introduction to Mobility	497
20.3.2 Towards a “Mobile CSP”	498
20.3.3 Pass the Port!	499
20.3.4 Closed Worlds	500
20.3.5 Opening Out	505
20.4 Notes	506
Notation	509
References	513
Index	519



<http://www.springer.com/978-1-84882-257-3>

Understanding Concurrent Systems

Roscoe, A.W.

2010, XVIII, 530 p., Hardcover

ISBN: 978-1-84882-257-3