

Contents

1. Overview	1
2. Context: Formal Methods in Software Engineering	5
2.1 The Place of Formal Methods in Software Engineering	5
2.2 The Role of Mathematics	6
2.3 Conditions for Using Inconsistencies Productively	7
2.4 Two Sides of Machine Support for Proofs	8
2.5 The Essence of Formal Methods in Software Engineering	9
2.6 Specific and General Formalisms	10
2.7 Goals and Consequences from the Analysis	12

Part I. Basic Concepts

3. Models of Time and of System Behaviors	15
3.1 Dense and Discrete Time Domains	15
3.2 Interval Sequences and Subclasses of Hybrid Systems	17
3.3 The Main Idea: Use of Infinitesimals	19
3.4 Summary	22
4. Infinitesimals	23
4.1 The Axiom of Idealization	24
4.2 The Axiom of Standardization	25
4.3 The Axiom of Transfer	25
4.4 More Structure Discerned in Classical Objects	26
4.5 Real-Time Systems with Constant Infinitesimal Steps	28
4.6 Summary	29
5. Operational Semantics of Discrete Systems	31
5.1 Action Systems	31
5.2 Abstract State Machines	32
5.2.1 Some Introductory Examples of ASM Rules	34
5.2.2 Terms	35
5.2.3 Rules	36
5.3 Effectivity	39
5.4 Classes of Symbols	40

5.5	Interaction with the Environment	42
5.6	Gurevich's Thesis	42
5.6.1	Elements of Programming Languages	43
5.6.2	Operationality	44
5.6.3	No Complications Induced by Formalism	44
5.7	Comparison to Other Formalisms for Discrete Systems	45
5.7.1	Updates vs. Transitions	45
5.7.2	State Based vs. Event Based Systems	46
5.7.3	Structured vs. Unstructured States	47
5.7.4	Explicit vs. Implicit Nondeterminism	47
5.7.5	Operationality vs. Declarativity	48
5.8	Summary	48
6.	Defining Hybrid Systems with ASMs	49
6.1	ASMs for the Definition of Classical Hybrid Systems	49
6.1.1	Standard Time ASM Rules and Hybrid Transition Systems	49
6.1.2	Infinite Activity	51
6.1.3	Hesitation and Urgency	52
6.2	ASMs with Infinitesimal Step Width	52
6.2.1	A Note on Zeno-ness in NTASMs	55
6.3	Simulation of an STASM by an NTASM	55
6.4	Well-Behaved Rules	58
6.5	Summary	62
7.	A Notation for a Temporal Logic	63
7.1	Semantic Domain	64
7.2	Interval Terms and Focused Predicates	64
7.3	Abbreviations	66
7.4	Examples of Valid Formulas	67
7.5	Fairness, Limited Activity and Other Example Specifications	68
7.6	On Accountability of a Step to Some Rule, and an Application to Synchronous Systems	69
7.7	Summary	72

Part II. Modelling Strategies

8.	Concurrency and Reactivity: Interleaving	75
8.1	The Interleaving Approach to Concurrency	76
8.2	Some Remarks on Fairness	78
8.3	Properties	79
8.4	Interleaving NTASM Models	80
8.5	On the Appropriateness of the Interleaving Abstraction	81
8.6	Summary	82

9. The Synchronous Approach to Concurrency	83
9.1 Reactive Systems as Mealy Automata	83
9.2 Composing I/O Automata	86
9.3 Micro-steps of Synchronous Systems as ASMs	90
9.4 Environment Interaction and the Synchrony Hypothesis	93
9.5 Synchronous NTASM Models	94
9.6 Summary	95
10. Deadlines	97
10.1 Synchronous NTASM Systems	98
10.2 Interleaving NTASM Systems	101
10.3 Admitting Infinitesimal Delays	103
10.4 Summary	106
11. Open Systems	107
11.1 Receptivity Simplified	107
11.2 (m,n)-Receptivity	109
11.3 Summary	112
12. Making Use of Different Magnitudes of Reals	113
12.1 The Magnitude Concept	114
12.2 Rule Schemes and the Ripple Counter Example	115
12.3 Making Delays Explicit	118
12.4 Analyzing a Logical Circuit for Hazards	121
12.5 Modelling Missing Knowledge Explicitly	124
12.6 Hazards Resulting from the Infinitesimal Discretization	126
12.7 Summary	127

Part III. Applications

13. A Case Study: Fischer's Protocol	131
13.1 A Hybrid Abstract State Machine Describing Fischer's Protocol	131
13.2 Specification and Proof of the Mutex Property	134
13.3 Infinitesimality of Step-Width and Plausibility of Assumptions	138
13.4 Summary	139
14. An ASM Meta-model for Petri Nets with Timing	141
14.1 ASM Models of Discrete Nets	141
14.2 Quantitatively Timed Nets	143
14.3 STASM Models of Doubly Timed Nets	145
14.3.1 An Interleaving Dynamics for Doubly Timed Nets ...	145
14.3.2 A Maximal Progress Dynamics for Doubly Timed Nets	147
14.3.3 Discussion of the STASM Models of Doubly Timed Nets	149

14.4	Comparison of STASM and NTASM Semantics	152
14.4.1	Well-Behavedness of the Interleaving Dynamics Rule for Doubly Timed Petri Nets	152
14.4.2	A Well-Behaved Rule for Interleaving Dynamics of Doubly Timed Petri Nets	155
14.5	Summary	159
15.	An ASM Meta-model for Timed and Hybrid Automata	161
15.1	An STASM Model of Hybrid Automata	162
15.2	Comments on the Modelling Choices	166
15.3	Timed Automata and Their Well-Behavedness	166
15.4	Well-Behavedness of Hybrid Automata	169
15.5	Summary	172
16.	A Production Cell with Timing	173
16.1	Introduction	173
16.2	Task Description	174
16.3	Requirements to Be Fulfilled by the Control Program	178
16.4	Direct Consequences from the Task Description	179
16.5	An Abstract Control Program	180
16.6	Schedules for Variable-Order Programs	183
16.7	One Crane, Order of Processing Units Fixed	183
16.8	Executing the Current Schedule	185
16.9	Two Cranes, Order of Processing Units Fixed	187
16.9.1	Splitting a Schedule into Segments	187
16.9.2	The Active and the Passive Crane and Their Tasks	188
16.9.3	Resting Position, Target Position and Initialization	189
16.9.4	Specifics of Crane Behavior	191
16.9.5	Waiting Times in a Two-Crane System	195
16.10	Are the System Properties Ensured?	199
16.11	Summary	201

Part IV. Summary

17.	Summary	205
A.	Common Notation	211
A.1	Non-standard Quantifiers and Predicates	211
A.2	Various Kinds of Expressions	211
A.3	Various Expressions for Functions and Sets of Functions	211
A.4	Some Common Sets	212
A.5	Some Definitions	212
References	215
Index	221

<http://www.springer.com/978-3-540-25576-5>

Operational Semantics for Timed Systems
A Non-standard Approach to Uniform Modeling of Timed
and Hybrid Systems

Rust, H.

2005, XII, 224 p., Softcover

ISBN: 978-3-540-25576-5