

# Table of Contents

Preface .....	V
---------------	---

---

## Part I. Discrete Event System Verification

---

### 1. Overview of Verification

Kenneth L. McMillan .....	3
1. Bugs .....	3
1.1 Concurrency bugs .....	3
1.2 An example .....	4
1.3 A real-life concurrency bug .....	5
2. The Mathematical Approach .....	5
2.1 Easy objections .....	6
2.2 The real objection .....	6
3. A Selective History .....	6
3.1 Early 1960's .....	6
3.2 Late 1960's .....	6
3.3 1970's .....	7
3.4 Late 1970's .....	8
3.5 1980's .....	10
3.6 Late 1980's .....	11
3.7 1990's .....	13

### 2. General Purpose Theorem Proving Methods in the Verification of Digital Hardware and Software

J. Strother Moore .....	14
1. Introduction .....	14
1.1 Formal models .....	15
1.2 Lessons .....	16
1.3 A spectrum of choices .....	17
1.4 Outline of this presentation .....	17
2. The ACL2 Theorem Prover .....	17
2.1 NQTHM: the Boyer-Moore system .....	17
2.2 A simple example of ACL2 .....	18

2.3	A more interesting example	21
2.4	Five key ideas in ACL2	26
3.	Using ACL2 to Model Digital Systems	26
4.	Industrial Scale Applications of ACL2	29
4.1	AMD floating point arithmetic	29
4.2	Motorola CAP digital signal processor	30
4.3	Summary of ACL2 applications	30
5.	Other General-Purpose Systems	31
5.1	HOL	31
5.2	PVS	32
6.	Conclusion	34
 <b>3. Temporal Logic and Model Checking</b>		
Kenneth L. McMillan		36
1.	Reactive Systems and Temporal Properties	36
1.1	Example: The alternating bit protocol	36
1.2	Temporal properties	36
1.3	Formalizing temporal properties	37
1.4	Model theory for temporal logic	39
1.5	Proofs in temporal logic	39
2.	Model Checking (Clarke/Emerson, Queille/Sifakis)	41
2.1	Example: Modeling a protocol in CSP (Hoare)	42
3.	Branching Time and CTL Model Checking	45
3.1	CTL model checking	47
3.2	Example: The ABP revisited	48
4.	Expressiveness Issues	52
4.1	Linear <i>vs.</i> branching time	52
4.2	Data independence	52
5.	Summary	53
 <b>4. Model Checking Using Automata Theory</b>		
Doron Peled		55
1.	$\omega$ -Automata	55
2.	Specification Using $\omega$ -Automata	57
3.	Operations on Büchi Automata	59
4.	Checking Emptiness	62
5.	Other Acceptance Conditions	63
6.	Translating LTL into Automata	64
6.1	Linear temporal logic	64
6.2	The translation algorithm	65
6.3	Improvements to the algorithm	74
7.	The Expressive Power of Büchi Automata	75
8.	The Complexity of LTL Model Checking	77

**5. Complexity Issues in Automata Theoretic Verification**

Sandeep K. Shukla .....	80
1. Introduction.....	80
1.1 About $\omega$ -automata based verification .....	81
2. About the COSPAN Verification Tool .....	84
2.1 Introduction.....	84
2.2 Modeling hardware.....	88
2.3 Specification and proof .....	96
2.4 Handling the complexity theoretic lowerbounds .....	97
3. The Theoretical Foundations: Boolean Algebra, Languages, and Selection-Resolution .....	98
3.1 Boolean algebra, automata and languages .....	98
3.2 Some words about Boolean algebras .....	99
3.3 L-automaton and L-process .....	103
3.4 Selection/resolution model .....	107
3.5 Verification based on L-processes and L-automata .....	108
4. Reduction Methodologies .....	110
4.1 Homomorphic reduction and refinement based topdown methodology .....	113
4.2 Inductive abstraction .....	115

**6. Symbolic Model Checking**

Kenneth L. McMillan.....	117
1. Introduction.....	117
2. Binary Decision Diagrams.....	117
2.1 Apply algorithm .....	119
2.2 The quantification algorithm .....	119
2.3 Circuit width and OBDD size .....	120
2.4 Variable ordering .....	121
3. Representing Sets and Relations .....	121
3.1 Characteristic functions of sets .....	121
3.2 Characteristic functions of relations .....	122
3.3 Forward and reverse image .....	123
3.4 Reachability analysis using OBDD's .....	124
4. Fixed Point Characterization of CTL .....	124
4.1 Fixed points of monotonic functions .....	125
4.2 Characterization of EG .....	126
4.3 Complexity of OBDD based model checking .....	126
5. The SMV System .....	127
5.1 SMV language.....	127
5.2 Example – a synchronous arbiter circuit .....	128
5.3 Example – distributed cache coherence protocol.....	130
6. The Mu-calculus and Symbolic Model Checking .....	132
6.1 Propositional mu-calculus .....	133

6.2 Mu-calculus and CTL .....	134
6.3 Relational mu-calculus and symbolic model checking .....	135
7. Summary .....	136
<b>7. Compositional Systems and Methods</b>	
Kenneth L. McMillan.....	138
1. Introduction.....	138
2. A Framework for Compositional Minimization .....	139
2.1 Framework .....	139
2.2 An example framework .....	140
2.3 Application: Decoupled processor controller .....	144
2.4 Hierarchical minimization .....	146
3. Assume/Guarantee Style Reasoning .....	147
3.1 Framework .....	147
3.2 An example framework .....	148
3.3 Why not ordinary CTL? .....	149
3.4 Application example – CPU controller revisited .....	149
4. Conclusion .....	151
<b>8. Symmetry and Model Checking</b>	
Kenneth L. McMillan.....	152
1. Symmetry and Model Checking .....	152
1.1 Permutations .....	152
1.2 Permutation groups .....	152
1.3 Symmetry in Kripke models .....	153
1.4 Reduced models .....	154
1.5 Checking CTL* formulas.....	156
2. Murphi – A Practical Approach to Symmetry Reductions .....	156
2.1 The Murphi language.....	157
2.2 Scalar sets .....	158
2.3 Cache protocol example.....	159
2.4 Data saturation.....	161
3. Summary .....	161
<b>9. Partial Order Reductions</b>	
Doron Peled.....	163
1. Introduction.....	163
2. Modeling Concurrent Systems .....	164
2.1 State spaces of concurrent systems .....	164
3. Stuttering Equivalence.....	165
3.1 Syntax and semantics of CTL*, CTL and LTL .....	166
4. Verification Using Representatives.....	167
4.1 Ample sub-state-spaces .....	167

5. Partial Order Reduction for Linear Specifications .....	168
5.1 The ample-sets reduction method .....	168
6. Reduction for Branching TL and Process Algebras .....	170
6.1 Behavioral equivalences .....	171
6.2 Correctness of the algorithm.....	173
7. Implementation Issues .....	175
8. Reducing the Visibility Constraint .....	178
9. Static Partial Order Reduction .....	179

## **10. Probabilistic Model Checking: Formalisms and Algorithms for Discrete and Real-time Systems**

Costas Courcoubetis and Stavros Tripakis .....	183
--	-----

1. Introduction.....	183
2. Preliminaries .....	187
2.1 Stochastic processes .....	187
3. Description Formalisms .....	187
3.1 Discrete-time probabilistic systems .....	187
3.2 Discrete-time probabilistic specifications .....	190
4. Complexity Results .....	192
4.1 Verification problems .....	192
4.2 Results for Markov chains .....	193
4.3 Results for concurrent Markov chains .....	193
5. Algorithms .....	193
5.1 Computing satisfaction probabilities for Markov chains .....	194
5.2 Checking emptiness for concurrent Markov chains .....	197
6. Extensions for ETL and PCTL* .....	199
6.1 Extended temporal logic .....	199
6.2 Probabilistic computation tree logic .....	200
7. Description Formalisms .....	201
7.1 Real-time probabilistic systems .....	201
7.2 Real-time probabilistic specifications .....	205
8. Complexity Results .....	208
8.1 Verification problems .....	208
8.2 Results .....	209
9. Algorithms .....	209
9.1 Region graph .....	209
9.2 TCTL algorithm .....	213
9.3 DTMA algorithm .....	215
10. Conclusions .....	217

## **11. Formal Verification in a Commercial Setting**

R. P. Kurshan .....	220
---------------------	-----

1. Introduction.....	220
2. Paradigms .....	222

3. Reduction .....	224
4. Interfaces .....	225
5. Support .....	227
6. Examples of Practice .....	227
7. Future .....	229

---

## Part II. Hybrid Systems: Modeling and Verification

---

### 12. Timed Automata

Rajeev Alur .....	233
1. Modeling .....	233
2. Reachability Analysis .....	240
3. Automata-Theoretic Verification .....	250
3.1 Verification via Automata Emptiness .....	250
3.2 Theory of Timed Languages .....	255
4. Tools and Applications .....	259
5. Discussion .....	260

### 13. The Theory of Hybrid Automata

Thomas A. Henzinger .....	265
1. Hybrid Automata .....	265
1.1 Syntax .....	265
1.2 Safe Semantics .....	267
1.3 Live Semantics .....	268
1.4 Composition .....	269
2. On the Trace Languages of Hybrid Automata .....	272
2.1 Verification Tasks .....	272
2.2 Rectangular Automata .....	273
2.3 Verification Results .....	275
3. On the State Spaces of Hybrid Automata .....	276
3.1 Symbolic Analysis of Transition Systems .....	276
3.2 Linear Hybrid Automata .....	280
3.3 Bisimilarity and Similarity Relations .....	282
3.4 Computation Tree Logics .....	285

### 14. On the Composition of Hybrid Systems

Sébastien Bornot and Joseph Sifakis .....	293
1. Introduction .....	293
2. Hybrid extensions of discrete systems .....	296
2.1 Discrete systems .....	296
2.2 Hybrid extension of $S_A$ .....	297
2.3 Comparing hybrid actions .....	298
3. Choice operators .....	302

3.1	Priority choice . . . . .	303
3.2	Consensual choice . . . . .	307
4.	Parallel composition . . . . .	310
4.1	Extending parallel composition from untimed to hybrid systems	310
4.2	Synchronization modes of hybrid actions . . . . .	313
5.	Applications . . . . .	319
6.	Discussion . . . . .	320

## **15. Reach Set Computation Using Optimal Control**

Pravin Varaiya . . . . .	323
1. Introduction . . . . .	323
2. Convex Reach Set Function . . . . .	324
3. Maximum principle . . . . .	328
4. Concluding remarks . . . . .	330

## **16. Control for a Class of Hybrid Systems**

Jan H. van Schuppen . . . . .	332
1. Introduction . . . . .	332
2. Example of Conveyor Belts . . . . .	333
3. Modeling of hybrid systems . . . . .	337
3.1 Definitions . . . . .	337
3.2 Subclasses of the Class of Hybrid Control Systems . . . . .	340
3.3 Realization of Hybrid Systems . . . . .	341
4. Control of Hybrid Systems . . . . .	342
4.1 Problem Formulation . . . . .	342
4.2 Example . . . . .	343
4.3 Control Synthesis for a Special Class of Systems . . . . .	346
4.4 Reachability Problems . . . . .	350
5. Concluding Remarks . . . . .	352

## **17. The SHIFT Programming Language and Run-time System for Dynamic Networks of Hybrid Automata**

Akash Deshpande, Aleks Göllü and Luigi Semenzato	355
1. Introduction	355
1.1 Related Work	356
2. The SHIFT Language	357
3. The SHIFT Model	360
3.1 Type Description	361
3.2 Component Description	362
3.3 World Description	363
3.4 World Semantics	364
4. The Particle Example	365
4.1 The Particle Type	365

4.2	The Source Type .....	367
4.3	The Monitor Type .....	368
4.4	Global Variables .....	369
5.	Conclusion .....	369

## **18. The Teja System for Real-Time Dynamic**

### **Event Management**

Akash Deshpande .....	372
1. Introduction.....	372
1.1 Enterprise Systems .....	372
1.2 Embedded Systems.....	373
1.3 Integrated Management.....	373
1.4 Performance.....	374
1.5 Tools Interfaces .....	374
2. The Teja Model .....	374
2.1 Basic Concepts .....	374
2.2 The Component Model .....	375
2.3 Event Model .....	378
2.4 Alert Model .....	378
2.5 Inheritance and Other Object-Oriented Features .....	378
2.6 Server Behavior .....	379
2.7 Client Behavior .....	380
3. Case Study: Real-time Electronic Funds Transfer.....	380
3.1 The TransferProcessor Component .....	381

## **19. Automated Highway Systems: an Example of Hierarchical Control**

Pravin Varaiya .....	391
1. Background .....	391
2. AHS design space .....	392
3. A control architecture .....	395
4. Design and verification of control .....	397
5. Remarks on layered control architectures .....	399

<b>Index .....</b>	<b>403</b>
--------------------	------------



<http://www.springer.com/978-3-540-65595-4>

Verification of Digital and Hybrid Systems

Inan, M.K.; Kurshan, R.P. (Eds.)

2000, XVIII, 405 p., Hardcover

ISBN: 978-3-540-65595-4