

Contents

1	Introduction	13
1.1	The Challenge of Randomization	13
1.1.1	Modeling	14
1.1.2	Verification	15
1.2	Organization of the Thesis	18
1.3	Reading the Thesis	22
2	An Overview of Related Work	23
2.1	Reactive, Generative and Stratified Models	23
2.1.1	Reactive Model	24
2.1.2	Generative and Stratified Models	25
2.2	Models based on Testing	26
2.3	Models with Nondeterminism and Denotational Models	28
2.3.1	Transitions with Sets of Probabilities	28
2.3.2	Alternating Models	28
2.3.3	Denotational Semantics	28
2.4	Models with Real Time	29
2.5	Verification: Qualitative and Quantitative Methods	29
2.5.1	Qualitative Method: Proof Techniques	29
2.5.2	Qualitative Method: Model Checking	30
2.5.3	Quantitative Method: Model Checking	31
3	Preliminaries	33
3.1	Probability Theory	33
3.1.1	Measurable Spaces	33
3.1.2	Probability Measures and Probability Spaces	33
3.1.3	Extensions of a Measure	34
3.1.4	Measurable Functions	34
3.1.5	Induced Measures and Induced Measure Spaces	35
3.1.6	Product of Measure Spaces	35
3.1.7	Combination of Discrete Probability Spaces	35
3.1.8	Conditional Probability	36
3.1.9	Expected Values	36
3.1.10	Notation	37
3.2	Labeled Transition Systems	37

3.2.1	Automata	37
3.2.2	Executions	39
3.2.3	Traces	40
3.2.4	Trace Semantics	40
3.2.5	Parallel Composition	40
4	Probabilistic Automata	43
4.1	What we Need to Model	43
4.2	The Basic Model	46
4.2.1	Probabilistic Automata	46
4.2.2	Combined Transitions	47
4.2.3	Probabilistic Executions	48
4.2.4	Notational Conventions	51
4.2.5	Events	52
4.2.6	Finite Probabilistic Executions, Prefixes, Conditionals, and Suffixes . . .	55
4.2.7	Notation for Transitions	58
4.3	Parallel Composition	61
4.3.1	Parallel Composition of Simple Probabilistic Automata	61
4.3.2	Projection of Probabilistic Executions	62
4.3.3	Parallel Composition for General Probabilistic Automata	70
4.4	Other Useful Operators	72
4.4.1	Action Renaming	72
4.4.2	Action Hiding	73
4.5	Discussion	73
5	Direct Verification: Stating a Property	75
5.1	The Method of Analysis	75
5.2	Adversaries and Adversary Schemas	79
5.2.1	Application of an Adversary to a Finite Execution Fragment	80
5.2.2	Application of an Adversary to a Finite Probabilistic Execution Fragment	80
5.3	Event Schemas	82
5.3.1	Concatenation of Event Schemas	82
5.3.2	Execution-Based Event Schemas	83
5.4	Probabilistic Statements	84
5.4.1	The Concatenation Theorem	84
5.5	Progress Statements	85
5.5.1	Progress Statements with States	86
5.5.2	Finite History Insensitivity	86
5.5.3	The Concatenation Theorem	87
5.5.4	Progress Statements with Actions	88
5.5.5	Progress Statements with Probability 1	89
5.6	Adversaries with Restricted Power	90
5.6.1	Execution-Based Adversary Schemas	91
5.6.2	Adversaries with Partial On-Line Information	91
5.7	Deterministic versus Randomized Adversaries	92

5.7.1	Execution-Based Adversary Schemas	93
5.7.2	Execution-Based Adversary Schemas with Partial On-Line Information	99
5.8	Probabilistic Statements without Adversaries	99
5.9	Discussion	100
6	Direct Verification: Proving a Property	103
6.1	How to Prove the Validity of a Probabilistic Statement	103
6.2	Some Simple Coin Lemmas	104
6.2.1	First Occurrence of an Action	106
6.2.2	First Occurrence of an Action among Many	107
6.2.3	I-th Occurrence of an Action among Many	109
6.2.4	Conjunction of Separate Coin Events	109
6.3	Example: Randomized Dining Philosophers	111
6.3.1	The Problem	111
6.3.2	The Algorithm	112
6.3.3	The High Level Proof	114
6.3.4	The Low Level Proof	116
6.4	General Coin Lemmas	121
6.4.1	Conjunction of Separate Coin Events with Multiple Outcomes	121
6.4.2	A Generalized Coin Lemma	124
6.5	Example: Randomized Agreement with Stopping Faults	126
6.5.1	The Problem	126
6.5.2	The Algorithm	127
6.5.3	The High Level Proof	128
6.5.4	The Low Level Proof	129
6.6	Example: The Toy Resource Allocation Protocol	130
6.7	The Partition Technique	132
6.8	Discussion	133
7	Hierarchical Verification: Trace Distributions	135
7.1	Introduction	135
7.1.1	Observational Semantics	135
7.1.2	Substitutivity and Compositionality	136
7.1.3	The Objective of this Chapter	137
7.2	Trace Distributions	138
7.3	Trace Distribution Preorder	141
7.4	Trace Distribution Precongruence	143
7.5	Alternative Characterizations of the Trace Distribution Precongruence	145
7.5.1	The Principal Context	145
7.5.2	High Level Proof	146
7.5.3	Detailed Proof	147
7.6	Discussion	165

8	Hierarchical Verification: Simulations	167
8.1	Introduction	167
8.2	Strong Simulations	167
8.3	Strong Probabilistic Simulations	171
8.4	Weak Probabilistic Simulations	172
8.5	Probabilistic Forward Simulations	172
8.6	The Execution Correspondence Theorem	176
8.6.1	Fringes	177
8.6.2	Execution Correspondence Structure	177
8.6.3	The Main Theorem	179
8.6.4	Transitivity of Probabilistic Forward Simulations	189
8.7	Probabilistic Forward Simulations and Trace Distributions	193
8.8	Discussion	194
9	Probabilistic Timed Automata	195
9.1	Adding Time	195
9.2	The Timed Model	196
9.2.1	Probabilistic Timed Automata	196
9.2.2	Timed Executions	198
9.3	Probabilistic Timed Executions	200
9.3.1	Probabilistic Time-Enriched Executions	201
9.3.2	Probabilistic Timed Executions	204
9.3.3	Probabilistic Executions versus Probabilistic Timed Executions	209
9.4	Moves	217
9.5	Parallel Composition	218
9.6	Discussion	222
10	Direct Verification: Time Complexity	223
10.1	General Considerations About Time	223
10.2	Adversaries	224
10.3	Event Schemas	224
10.4	Timed Progress Statements	226
10.5	Time Complexity	226
10.5.1	Expected Time of Success	227
10.5.2	From Timed Progress Statements to Expected Times	227
10.6	Example: Randomized Dining Philosophers	232
10.6.1	Representation of the Algorithm	232
10.6.2	The High Level Proof	233
10.6.3	The Low Level Proof	233
10.7	Abstract Complexity Measures	238
10.8	Example: Randomized Agreement with Time	240
10.9	Discussion	242

11 Hierarchical Verification: Timed Trace Distributions	243
11.1 Introduction	243
11.2 Timed Traces	243
11.3 Timed Trace Distributions	246
11.3.1 Three ways to Define Timed Trace Distributions	246
11.3.2 Timed Trace Distribution of a Trace Distribution	248
11.3.3 Action Restriction	249
11.4 Timed Trace Distribution Precongruence	249
11.5 Alternative Characterizations	250
12 Hierarchical Verification: Timed Simulations	257
12.1 Introduction	257
12.2 Probabilistic Timed Simulations	257
12.3 Probabilistic Timed Forward Simulations	258
12.4 The Execution Correspondence Theorem: Timed Version	259
12.4.1 Timed Execution Correspondence Structure	259
12.4.2 The Main Theorem	260
12.4.3 Transitivity of Probabilistic Timed Forward Simulations	260
12.5 Soundness for Timed Trace Distributions	260
13 Conclusion	263
13.1 Have we Met the Challenge?	263
13.2 The Challenge Continues	264
13.2.1 Discrete versus Continuous Distributions	264
13.2.2 Simplified Models	264
13.2.3 Beyond Simple Probabilistic Automata	265
13.2.4 Completeness of the Simulation Method	266
13.2.5 Testing Probabilistic Automata	266
13.2.6 Liveness in Probabilistic Automata	266
13.2.7 Temporal Logics for Probabilistic Systems	267
13.2.8 More Algorithms to Verify	267
13.2.9 Automatic Verification of Randomized Systems	268
13.3 The Conclusion's Conclusion	268
Bibliography	269
Table of Symbols	277

