

---

# Contents

<b>1</b>	<b>Introduction</b> .....	1
1.1	Formal Methods in System Design .....	1
1.1.1	General Remarks and Taxonomy .....	1
1.1.2	Classification of Formal Methods .....	4
1.1.3	Classification of Systems .....	10
1.2	Genealogy of Formal Verification .....	16
1.2.1	Early Beginnings of Mathematical Logic .....	16
1.2.2	Automated Theorem Proving .....	20
1.2.3	Beginnings of Program Verification .....	23
1.2.4	Dynamic Logics and Fixpoint Calculi .....	24
1.2.5	Temporal Logics .....	28
1.2.6	Decidable Theories and $\omega$ -Automata .....	33
1.2.7	Summary .....	38
1.3	Outline of the Book .....	40
<b>2</b>	<b>A Unified Specification Language</b> .....	45
2.1	Kripke Structures as Formal Models of Reactive Systems .....	45
2.1.1	Simulation and Bisimulation of Kripke Structures .....	53
2.1.2	Quotient Structures .....	61
2.1.3	Products of Kripke Structures .....	66
2.2	Syntax of the Specification Logic $\mathcal{L}_{\text{spec}}$ .....	68
2.3	Semantics of the Specification Logic $\mathcal{L}_{\text{spec}}$ .....	77
2.4	Normal Forms .....	84
<b>3</b>	<b>Fixpoint Calculi</b> .....	89
3.1	Partial Orders, Lattices and Fixpoints .....	90
3.2	The Basic $\mu$ -Calculus .....	98
3.3	Monotonicity of State Transformers .....	103
3.4	Model Checking of the Basic $\mu$ -Calculus .....	108
3.4.1	A Naive Model Checking Procedure .....	108
3.4.2	Optimization by the Alternation Depth .....	111

3.5	Vectorized $\mu$ -Calculus	118
3.5.1	State Transformers of Vectorized Fixpoint Expressions	119
3.5.2	Decomposing Equation Systems	124
3.5.3	Model Checking Vectorized Fixpoint Expressions	129
3.5.4	Comparing Basic and Vectorized $\mu$ -Calculus Model Checking	138
3.5.5	Dependency-Triggered Evaluations	142
3.5.6	The Cleaveland-Steffen Algorithm	148
3.6	Reducing the Alternation Depth w.r.t. Structures	159
3.7	Computing Fair States	164
3.8	Final Remarks on Completeness and Expressiveness	169
3.8.1	Bisimilarity and the Future Fragment	169
3.8.2	Relationship to $\omega$ -Tree Automata and Games	173
3.8.3	Dynamic Logic	175
<b>4</b>	<b>Finite Automata</b>	183
4.1	Regular Languages, Regular Expressions and Automata	186
4.2	The Logic of Automaton Formulas	189
4.3	Boolean Closure	194
4.4	Converting Automaton Classes	202
4.5	Determinization and Complementation	209
4.5.1	The Rabin-Scott Subset Construction	210
4.5.2	Determinization of $\text{NDet}_F$	213
4.5.3	Determinization of $\text{NDet}_G$	215
4.5.4	Determinization of $\text{NDet}_{FG}$	219
4.5.5	Reducing $\text{NDet}_{GF}$ to $\text{Det}_{\text{Rabin}}$	223
4.6	The Hierarchy of $\omega$ -Automata and the Borel Hierarchy	236
4.7	Automata and Monoids	252
4.7.1	Finite Semigroups and Monoids	252
4.7.2	Automata and Their Monoids	257
4.8	Decision Procedures for $\omega$ -Automata	264
4.8.1	Flattening $\omega$ -Automata	265
4.8.2	Translating $\mathcal{L}_\omega$ Model Checking to $\mathcal{L}_\mu$ Model Checking	267
4.8.3	Translating Automata to Vectorized $\mu$ -Calculus	270
<b>5</b>	<b>Temporal Logics</b>	279
5.1	Introduction	279
5.2	Branching Time Logics – Sublanguages of $\text{CTL}^*$	284
5.2.1	$\text{CTL}$ , $\text{LTL}$ and $\text{CTL}^*$	285
5.2.2	Adding Syntactic Sugar to $\text{CTL}$	292
5.3	Translating Temporal Logics to the $\mu$ -Calculus	299
5.3.1	$\text{CTL}$ and $\text{FairCTL}$ as Fragments of the $\mu$ -Calculus	300
5.3.2	$\text{CTL}^2$ as a Fragment of the $\mu$ -Calculus	302
5.3.3	Eliminating Quantified Boolean Expressions	304

5.3.4	Adding Path Quantifiers	310
5.3.5	Translating LeftCTL* to Vectorized $\mu$ -Calculus	313
5.4	Translating Temporal Logics to $\omega$ -Automata	329
5.4.1	The Basic Translation from $LTL_p$ to $NDet_{Streett}$	331
5.4.2	Exploitation of Monotonicity	343
5.4.3	Borel Classes of Temporal Logic	348
5.4.4	Reducing Temporal Borel Classes to Borel Automata	355
5.4.5	Reductions to CTL/LeftCTL* Model Checking	365
5.5	Completeness and Expressiveness of Temporal Logic	375
5.5.1	Noncounting Automata and Temporal Logic	376
5.5.2	Completeness of the Borel Classes	383
5.5.3	Completeness of the Future Fragments	387
5.6	Complexities of the Model Checking Problems	393
5.7	Reductions by Simulation and Bisimulation Relations	400
<b>6</b>	<b>Predicate Logic</b>	<b>405</b>
6.1	Introduction	405
6.2	Predicate Logics	408
6.2.1	Syntax and Semantics	408
6.2.2	Basics of Predicate Logic	410
6.2.3	Fragments with Decidable Satisfiability Problem	415
6.2.4	Embedding Modal Logics in Predicate Logic	421
6.2.5	Predicate Logic on Linearly Ordered Domains (on $\mathbb{N}$ )	424
6.3	Monadic Second Order Logic of Linear Order $MSO_{<}$	428
6.3.1	Equivalence of S1S and $MSO_{<}$	428
6.3.2	Translating $MSO_{<}$ to $\omega$ -Automata	434
6.3.3	Büchi's Decision Procedure: Normal Forms for S1S	439
6.4	Monadic First Order Logic of Linear Order $MFO_{<}$	442
6.5	Non-Monadic Characterizations	452
<b>7</b>	<b>Conclusions</b>	<b>455</b>
<b>A</b>	<b>Binary Decision Diagrams</b>	<b>459</b>
A.1	Basic Definitions	459
A.2	Basic Algorithms on BDDs	466
A.3	Minimization of BDDs Using Care Sets	471
A.4	Computing Successors and Predecessors	477
A.5	Variable Reordering	483
A.6	Final Remarks	486
<b>B</b>	<b>Local Model Checking and Satisfiability Checking for the <math>\mu</math>-Calculus</b>	<b>487</b>
B.1	A Partial Local Model Checking Procedure	488
B.2	A Complete Local Model Checking Procedure	493
B.3	Satisfiability of $\mu$ -Calculus Formulas	500

<b>C</b>	<b>Reduction of Structures</b> .....	527
C.1	Galois Connections and Simulations .....	527
C.1.1	Basic Properties of Galois Connections .....	528
C.1.2	Galois Simulation .....	531
C.2	Abstract Structures and Preservation Results .....	534
C.3	Optimal and Faithful Abstractions .....	537
C.4	Data Abstraction .....	542
C.4.1	Abstract Interpretation of Structures .....	544
C.4.2	Abstract Specifications .....	549
C.5	Symmetry and Model Checking .....	551
C.5.1	Symmetries of Structures .....	552
C.5.2	Symmetries in the Specification .....	557
	<b>References</b> .....	561
	<b>Index</b> .....	591