

Wojciech Penczek, Agata Półrola

Advances in Verification of Time Petri Nets and Timed Automata

Studies in Computational Intelligence, Volume 20

Editor-in-chief

Prof. Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
ul. Newelska 6
01-447 Warsaw
Poland
E-mail: kacprzyk@ibspan.waw.pl

Further volumes of this series
can be found on our homepage:
springer.com

Vol. 5. Da Ruan, Guoqing Chen, Etienne E.
Kerre, Geert Wets (Eds.)
Intelligent Data Mining, 2005
ISBN 3-540-26256-3

Vol. 6. Tsau Young Lin, Setsuo Ohsuga,
Churn-Jung Liao, Xiaohua Hu, Shusaku
Tsumoto (Eds.)
*Foundations of Data Mining and Knowledge
Discovery*, 2005
ISBN 3-540-26257-1

Vol. 7. Bruno Apolloni, Ashish Ghosh, Ferda
Alpaslan, Lakhmi C. Jain, Srikanta Patnaik
(Eds.)
Machine Learning and Robot Perception,
2005
ISBN 3-540-26549-X

Vol. 8. Srikanta Patnaik, Lakhmi C. Jain,
Spyros G. Tzafestas, Germano Resconi,
Amit Konar (Eds.)
Innovations in Robot Mobility and Control,
2005
ISBN 3-540-26892-8

Vol. 9. Tsau Young Lin, Setsuo Ohsuga,
Churn-Jung Liao, Xiaohua Hu (Eds.)
*Foundations and Novel Approaches in Data
Mining*, 2005
ISBN 3-540-28315-3

Vol. 10. Andrzej P. Wierzbicki, Yoshiteru
Nakamori
Creative Space, 2005
ISBN 3-540-28458-3

Vol. 11. Antoni Ligęza
*Logical Foundations for Rule-Based
Systems*, 2006
ISBN 3-540-29117-2

Vol. 12. Jonathan Lawry
*Modelling and Reasoning with Vague Con-
cepts*, 2006
ISBN 0-387-29056-7

Vol. 13. Nadia Nedjah, Ajith Abraham,
Luiza de Macedo Mourelle (Eds.)
Genetic Systems Programming, 2006
ISBN 3-540-29849-5

Vol. 14. Spiros Sirmakessis (Ed.)
Adaptive and Personalized Semantic Web,
2006
ISBN 3-540-30605-6

Vol. 15. Lei Zhi Chen, Sing Kiong Nguang,
Xiao Dong Chen
*Modelling and Optimization of
Biotechnological Processes*, 2006
ISBN 3-540-30634-X

Vol. 16. Yaochu Jin (Ed.)
Multi-Objective Machine Learning, 2006
ISBN 3-540-30676-5

Vol. 17. Te-Ming Huang, Vojislav Kecman,
Ivica Koprija
*Kernel Based Algorithms for Mining Huge
Data Sets*, 2006
ISBN 3-540-31681-7

Vol. 18. Chang Wook Ahn
Advances in Evolutionary Algorithms, 2006
ISBN 3-540-31758-9

Vol. 19. Ajita Ichalkaranje, Nikhil
Ichalkaranje, Lakhmi C. Jain (Eds.)
*Intelligent Paradigms for Assistive and
Preventive Healthcare*, 2006
ISBN 3-540-31762-7

Vol. 20. Wojciech Penczek, Agata Pórola
*Advances in Verification of Time Petri Nets
and Timed Automata*, 2006
ISBN 3-540-32869-6

Wojciech Penczek
Agata Pólróla

Advances in Verification of Time Petri Nets and Timed Automata

A Temporal Logic Approach

With 124 Figures

 Springer

Doc. dr. hab. Wojciech Penczek
Institute of Computer Science
Polish Academy of Sciences
ul. Ordona 21
01-237 Warsaw
Poland
and

Institute of Informatics
Podlasie Academy
ul. Sienkiewicza 51
08-110 Siedlce
Poland
E-mail: penczek@ipipan.waw.pl

Dr. Agata Półrola
Faculty of Mathematics
University of Łódź
ul. Banacha 22
90-238 Łódź
Poland
E-mail: polrola@math.uni.lodz.pl

Library of Congress Control Number: 2006925476

ISSN print edition: 1860-949X

ISSN electronic edition: 1860-9503

ISBN-10 3-540-32869-6 Springer Berlin Heidelberg New York

ISBN-13 978-3-540-32869-8 Springer Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable to prosecution under the German Copyright Law.

Springer is a part of Springer Science+Business Media
springer.com
© Springer-Verlag Berlin Heidelberg 2006
Printed in the Netherlands

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover design: deblik, Berlin

Typesetting by the authors and SPI Publisher Services

Printed on acid-free paper SPIN: 11599357

89/SPI

5 4 3 2 1 0

To our families

Introduction

Verification of real-time systems is an important subject of research. This is highly motivated by an increasing demand to verify safety critical systems, i.e., time-dependent distributed systems, failure of which could cause dramatic consequences for both people and hardware.

Temporal logic methods have been used for verification over the last twenty years, proving their usefulness for such an application. Whereas infinite state systems still require deductive proof methods, systems of finite abstract models can be verified using algorithmic approaches. This means that the verification process can be fully automated. One of the most promising sets of techniques for verification is known as *model checking*. Essentially, in this formalism verifying that a property follows from a system specification amounts to checking whether or not a temporal formula is valid on a model representing all the possible computations of the system.

Several models of real-time systems are usually considered in the literature, but timed automata (TA) [10] and time Petri nets (TPNs) [106] belong to the most widely used. For these models, one is, usually, interested in checking reachability or more involved temporal properties that are typically expressed either in a standard temporal logic like LTL and CTL*, or in a timed extension of CTL, called TCTL [7]. Unfortunately, practical applicability of model checking methods is strongly limited by the *state explosion problem*, which makes models grow exponentially in the number of the concurrent processes of a system. For real-time systems, this problem occurs with a particular strength, which follows from infinity of the dense time domain. Therefore, existing verification techniques frequently apply symbolic representations of state spaces using either operations on Difference Bound Matrices [68] or similar structures [32] for representing states of abstract models, or variations of Boolean Decision Diagrams like Clock Decision Diagrams (CDDs) [24, 166], Numeric Decision Diagrams (NDDs) [18], Difference Decision Diagrams (DDD) [107, 108], or Clock Restriction Diagrams (CRDs) [168].

Quite recently, a new approach to verification of real time systems, based on SAT-related algorithms, has been suggested. The reason for this is a dra-

matic increase in efficiency of SAT-solvers over the last few years. The SAT-based approach can exploit either a sequence of translations starting from a timed system and a timed temporal property, going via (quantified) separation logic to quantified propositional logic and further to propositional logic [20, 110, 140] or a direct translation from a timed system and a timed temporal property to propositional logic [120, 171, 177].

Finite state models for timed systems, preserving properties to be checked, are usually built using the detailed region graph approach or (possibly minimal) abstract models, based on state classes or regions. Algorithms for generating such models have been defined for time Petri nets [32, 35, 73, 102, 111, 117, 163, 174], as well as for timed automata [7, 8, 41, 66, 125, 159].

It seems that in spite of the same underlying timed structure, model checking methods for time Petri nets and timed automata have been developed independently of each other. However, several attempts to combine the two approaches were made, concerning both a structural translation of one model into the other [52, 60, 78, 89, 103, 124, 144] or an adaptation of existing verification techniques [73, 111, 163].

The aim of this monograph is to present a recent progress in the development of two model checking methods, based on either building abstract state spaces or application of SAT-based symbolic techniques. The latter is achieved indirectly for time Petri nets, namely via a translation to timed automata. Our special emphasis is put not only on the verification methods, but also on specification languages and their semantics.

Structure of the book

Chapter 1 of this book introduces Petri nets, discusses their time extensions, and provides a definition of time Petri nets. Our attention is focused on a special kind of TPNs – distributed time Petri nets, which, in a sense, correspond to networks of timed automata introduced in Chapter 2. Two main alternative approaches to the semantics of time Petri nets are considered. The first of them consists in assigning clocks to various components of the net, i.e., the transitions, the places, or the processes, whereas the second exploits the so-called *firing intervals* of the transitions. The chapter ends with comparing the above semantics as well as their dense and discrete versions.

Chapter 2 considers timed automata, which were introduced by Alur and Dill [10]. Timed automata are extensions of finite state automata. We give semantics of timed automata and show how to define their product. Typically, we consider *networks* of timed automata, consisting of several concurrent TA running in parallel and communicating with each other. Concrete models are defined and progressiveness is discussed.

Chapter 3 deals with various structural translations from TPNs to TA. They enable an application of specific verification methods for timed automata to time Petri nets. Several methods of translating time Petri nets to timed automata have been already developed. However, in most cases translations

produce automata which extend timed automata. We sketch some of the existing approaches, but focus mainly on the translations that correspond to the semantics of time Petri nets, associating clocks with various components of the nets, like the places, the transitions, or the processes.

Chapter 4 introduces temporal specification languages. We start our presentation with the standard branching time logic CTL^* , its extension modal μ -calculus, and then discuss timed temporal logics: TCTL and timed μ -calculus. It is important to mention that we consider two versions of syntax of TCTL interpreted over either weakly or strongly monotonic runs.

Chapter 5 gives model abstraction methods based on state classes approaches for TPNs and on partition refinement for TA. For time Petri nets we discuss different abstract models like state class graphs, geometric region graphs, atomic state class graphs, pseudo-atomic state class graphs, and strong state class graphs. For timed automata we concentrate on detailed region graphs, (pseudo-)bisimulating models, (pseudo-)simulating models, and forward-reachability graphs. The last section of this chapter gives an overview of difference bound matrices (DBMs), which are used for representing states of abstract models.

Chapter 6 deals with model checking methods for CTL. These methods include a standard state labelling algorithm as well as an automata-theoretic approach. Moreover, we show that model checking for TCTL over timed automata can be reduced to model checking for CTL.

Chapter 7 discusses SAT-based verification techniques, like bounded (BMC) and unbounded model checking (UMC). The main idea behind BMC [120, 171] consists in translating the model checking problem for an existential fragment of some branching-time temporal logic (like CTL or TCTL) on a fraction of a model into a test of propositional satisfiability, for which refined tools already exist [109]. Unlike BMC, UMC [105, 140] deals with unrestricted temporal logics checked on complete models at the price of a decrease in efficiency.

Each chapter of our book is accompanied with pointers to the literature, where descriptions of complementary methods or formalisms can be found.

Acknowledgement. The authors would like to thank the following people for commenting on this monograph: Bernard Berthomieu, Franck Cassez, Piotr Dembiński, Magdalena Kacprzak, Sławomir Lasota, Oded Maler, Olivier H. Roux, Maciej Szreter, Stavros Tripakis, Bożena Woźna, Tomohiro Yoneda and Andrzej Zbrzezny. All the comments from the above experts have greatly helped to improve the book.

The authors acknowledge support from Ministry of Science and Education (grant No. 3T11C 011 28).

Warsaw, Poland,
December 2005

Wojciech Penczek
Agata Pótróla

Contents

List of Figures	XV
List of Symbols	XXI

Part I Specifying Timed Systems and Their Properties

1 Petri Nets with Time	3
1.1 Incorporating Time into Petri Nets	6
1.2 Time Petri Nets	7
1.2.1 Distributed Time Petri Nets	8
1.2.2 Semantics: The Clock Approach	11
Clocks Assigned to the Transitions	11
Clocks Assigned to the Places	12
Clocks Assigned to the Processes	14
1.2.3 Semantics: Firing Intervals	15
1.3 Reasoning about Time Petri Nets	17
1.3.1 Comparison of the Semantics	17
1.3.2 Dense versus Discrete Semantics	18
Alternative (Discrete) Semantics	21
1.3.3 Concrete Models for TPNs	23
1.3.4 Progressiveness in Time Petri Nets	23
1.3.5 The Case of “General” TPNs	27
2 Timed Automata	29
2.1 Time Zones	29
2.2 Networks of Timed Automata	33
2.3 Semantics of Timed Automata	38
2.3.1 Alternative (Discrete) Semantics	40
2.4 Concrete Models for TA	41
2.5 Checking Progressiveness	42

2.5.1	A Static Technique	44
2.5.2	Applying Verification Methods	46
2.5.3	A Solution for Strongly Monotonic Semantics	48
3	From Time Petri Nets to Timed Automata	51
3.1	Translations to Extended Timed Automata	51
3.2	Translation for “Clocks Assigned to the Transitions”	53
3.3	Translation for “Clocks Assigned to the Places”	55
3.3.1	Supplementary Algorithms	57
	Obtaining General TA	57
	Obtaining Diagonal-Free TA	58
3.4	Translation for Distributed Nets	59
3.5	Comparing Expressiveness of TPNs and TA	62
4	Main Formalisms for Expressing Temporal Properties	63
4.1	Non-temporal Logics	63
4.1.1	Propositional Logic (PL)	63
	Syntax of PL	63
	Semantics of PL	64
4.1.2	Quantified Propositional Logic (QPL)	65
	Syntax of QPL	65
	Semantics of QPL	65
4.1.3	(Quantified) Separation Logic ((Q)SL)	65
	Syntax of (Q)SL	65
	Semantics of (Q)SL	66
4.2	Untimed Temporal Logics	66
4.2.1	Computation Tree Logic* (CTL*)	67
	Syntax of CTL*	67
	Sublogics of CTL*	67
	Semantics of CTL*	69
4.2.2	Modal μ -Calculus	71
	Syntax of Modal μ -Calculus	71
	Semantics of Modal μ -Calculus	72
4.2.3	Interpretation of Temporal Logics over Timed Systems Models	74
4.3	Timed Temporal Logics	75
4.3.1	Timed Computation Tree Logic (TCTL)	75
	Syntax of TCTL	75
	Semantics of TCTL over Timed Systems	76
	Strongly Monotonic Interval Semantics	76
	Weakly Monotonic Interval Semantics	78
4.3.2	Timed μ -Calculus ($T\mu$)	80
	Syntax of $T\mu$	80
	Semantics of $T\mu$	80
4.3.3	Syntax and Semantics of $TCTL_C$	82

Part II Model Generation and Verification

5	Abstract Models	89
5.1	Model Generation for Time Petri Nets	96
5.1.1	State Class Approaches	96
	State Class Graph	96
	Geometric Region Graph	103
	Atomic State Class Graph	107
	Pseudo-atomic State Class Graph	111
	Strong State Class Graph	112
	Strong Atomic State Class Graph	115
5.1.2	Other Approaches – an Overview	115
5.2	Model Generation for Timed Automata	115
5.2.1	Detailed Region Graphs	116
5.2.2	Partition Refinement	122
	Bisimulating Models	124
	The Algorithm of Bouajjani et al.	124
	A Convexity-Preserving Technique	127
	The Lee–Yannakakis Algorithm	130
	Simulating Models	133
	Pseudo-bisimulating Models	135
	A Convexity-Preserving Technique	137
	Pseudo-simulating Models	137
	Other Minimization Techniques	138
5.2.3	Forward-reachability Graphs	138
	Abstractions for Forward-reachability Graphs	140
	Inclusion Abstraction	140
	Extrapolation Abstraction	142
	Other Abstractions	145
5.3	Difference Bounds Matrices	146
5.3.1	Operations on Zones Using DBMs	148
	Intersection	148
	Clock Resets	148
	Time Successor	150
	Time Predecessor	150
	Immediate Time Predecessor	150
	Zone Difference	153
6	Explicit Verification	155
6.1	Model Checking for CTL	155
6.1.1	State Labelling	155
6.1.2	Automata-Theoretic Approach	158
	Alternating Automata	158
	Translation from CTL to WAA	164

	Model Checking with WAA	166
	Product Automaton	167
	Checking Non-emptiness of 1-Letter Word WAA	170
6.2	Model Checking for TCTL over Timed Automata	171
6.2.1	Verification Using Translation from TCTL to CTL	171
	Model Checking over Region Graph Models	172
	Model Checking over (Bi)simulating Models	178
6.2.2	Overview of Other Approaches to TCTL Verification ..	178
6.3	Selected Tools	178
6.3.1	Tools for TPNs	179
6.3.2	Tools for TA	179
7	Verification Based on Satisfiability Checking	181
7.1	Bounded Model Checking Using Direct Translation to SAT ...	182
7.1.1	Discretization of TA	182
	Discretized Model for (Un)reachability Verification	185
	Discretized Model for TCTL Verification	188
7.1.2	Bounded Model Checking for ECTL^r_X	191
7.1.3	Checking Reachability with BMC	201
7.1.4	Checking Unreachability with BMC	201
7.2	Bounded Model Checking via Translation to SL	203
7.2.1	Encoding k -Paths	203
7.2.2	Product Encoding	204
7.2.3	Encoding of LTL Formulas	205
	No loop case	205
	Loop case	207
7.2.4	Optimizations of the Encoding	207
7.3	Unbounded Model Checking for $T\mu$	207
7.3.1	From Real to Boolean Quantification	211
7.3.2	From QSL Formulas Without Real-Time Quantification to Equivalent Boolean Formulas	212
7.4	Deciding Separation Logic (MATH-SAT)	212
7.4.1	From SL to Propositional Logic	212
	Propositional Encoding	213
	Building the Inequality Graph	213
	Transitivity Constraints	214
	Complexity and Optimization	216
7.4.2	Other Approaches to Deciding SL	217
7.5	Deciding Propositional Formulas (SAT)	218
7.5.1	Boolean Constrain Propagation	220
7.5.2	Conflict-Based Learning	222
7.5.3	Variable Selection (VS)	224
7.6	From a Fragment of QPL to PL	225
7.7	Remaining Symbolic Approaches - Overview	229
7.8	Selected Tools	229

Concluding Remarks and Future Research Directions 231

References 233

Index 245

Author Index 251

List of Figures

1.1	A Petri net	4
1.2	A bounded Petri net	6
1.3	A time Petri net	8
1.4	A distributed time Petri net with disjoint processes	9
1.5	A distributed time Petri net with communicating processes (Fischer's mutual exclusion protocol for $n = 2$)	10
1.6	The time Petri net used in Example 1.13	21
1.7	A time Petri net with structural deadlock	24
1.8	A deadlock-free time Petri net with a structural deadlock.....	24
1.9	A time Petri net with deadlock (only non-zero time steps allowed).....	25
1.10	A time Petri net with a non-progressive run	26
2.1	Polyhedra in \mathbb{R}_{0+}^2	31
2.2	Zones $Z, Z' \in Z(2)$ and examples of operations on them	32
2.3	A timed automaton.....	34
2.4	Fischer's mutual exclusion protocol for two processes.....	34
2.5	Composition of timed automata	35
2.6	Train–Gate–Controller example	36
2.7	The product automaton for the Train–Gate–Controller example	37
2.8	A timed automaton with timelocks	42
2.9	A timed automaton with a deadlock in the strongly monotonic semantics	43
2.10	A timed automaton with strongly progressive structural loops..	44
2.11	The timed automaton considered in Example 2.17	45
2.12	The timed automaton considered in Example 2.19	47
2.13	A timed automaton for testing deadlocks in the TA of Fig. 2.12	48
2.14	The automaton \mathcal{A}_m used to test progressiveness of the automaton of Fig. 2.11 in the strongly monotonic semantics ...	49

XVIII List of Figures

3.1	The timed automaton for the net of Fig. 1.3 resulting from the translation for “clocks assigned to the transitions”	55
3.2	Partitioning $\llbracket inv_{\mathcal{N}}(m_{34}) \rrbracket$ into disjoint zones – the algorithm for non-diagonal-free TA	58
3.3	Partitioning $\llbracket inv_{\mathcal{N}}(m_{34}) \rrbracket$ into disjoint zones – the algorithm for diagonal-free TA	58
3.4	The timed automaton for the net in Fig. 1.3 resulting from the translation for “clocks assigned to the places”	59
3.5	The timed automaton for the net in Fig. 1.3 resulting from the translation for “clocks assigned to the processes”	61
4.1	Examples of CTL* formulas	70
4.2	Relations between sublogics of CTL*	71
4.3	A sequence of approximations for computing $E(\wp_1 U \wp_2)$	73
4.4	Examples of TCTL formulas	77
4.5	Weakly monotonic interval semantics for TCTL	79
4.6	Examples of $T\mu$ formulas	82
4.7	Some examples of TCTL _C formulas	84
5.1	The conditions EE, EA, AE and U in abstract state spaces	91
5.2	Conditions on transition relation in pseudo-bisimulating and pseudo-simulating models	93
5.3	Various abstract models built for the same underlying concrete one	94
5.4	Time-abstracted and discrete successor relations	95
5.5	The time Petri net considered in Example 5.3	98
5.6	An algorithm for generating state class graph of a time Petri net	100
5.7	The time Petri net considered in Example 5.6	101
5.8	Constructing the state class graph for the net in Fig. 5.7	101
5.9	Constructing the geometric region graph for the net in Fig. 5.7	105
5.10	The geometric region graph of the modified equivalence relation, built for the net in Fig. 5.7	107
5.11	An algorithm for generating atomic state class graph of a (1-safe) time Petri net	108
5.12	The atomic state class graph for the net in Fig. 5.7	111
5.13	The strong state class graph for the net in Fig. 5.7	114
5.14	Detailed zones for $c_{max}(\mathcal{A}, \varphi) = 1$	117
5.15	Time- and action successor relation for detailed regions	119
5.16	Time- and action successor relation for detailed regions when boundary regions are distinguished	120
5.17	The timed automaton of Example 5.14	121
5.18	The reachable part of the detailed region graph for the automaton of Fig. 5.17 and a formula φ s.t. $c_{max}(\mathcal{A}, \varphi) = 1$	121

5.19	The reachable part of the boundary-distinguishing detailed region graph for the automaton of Fig. 5.17 and a formula φ such that $c_{max}(\mathcal{A}, \varphi) = 1$	122
5.20	A generic minimization algorithm	124
5.21	Partitioning of the classes to preserve AE.....	125
5.22	A partitioning process for bisimulating models	126
5.23	A minimization algorithm for building strong ta-bisimulating models for TA without computing differences	128
5.24	The functions TimeSplit and ActionSplit	129
5.25	A generic minimization algorithm by Lee and Yannakakis	130
5.26	Partitioning of a class in the Lee-Yannakakis algorithm.....	132
5.27	Partitioning of the classes to preserve U	134
5.28	A minimization algorithm for testing reachability on pseudo-bisimulating models	136
5.29	A general reachability algorithm.....	139
5.30	The timed automaton used in Example 5.17	140
5.31	The forward-reachability graph for the automaton of Fig. 5.30 ..	140
5.32	A forward-reachability graph with the inclusion abstraction built for the automaton of Fig. 5.30.....	141
5.33	A timed automaton whose forward-reachability graph with the inclusion abstraction is infinite.....	141
5.34	Extrapolations of zones for $c_{max}(\mathcal{A}) = 2$ (diagonal-free TA)	142
5.35	A forward-reachability graph with the extrapolation abstraction built for the automaton of Fig. 5.30	143
5.36	A timed automaton with clock differences	144
5.37	The forward-reachability graph for the automaton of Fig. 5.36 ..	144
5.38	Convex hulls for zones Z_1, Z_2	146
5.39	A graphical interpretation of bounds of the canonical DBM	148
5.40	A non-canonical DBM for intersection of two zones represented by canonical DBMs	149
5.41	The steps of computing $[\{x_1\} := 0]Z$	149
5.42	(a) A correct time-successor of the zone represented by a canonical DBM, (b) an incorrect time-successor of the zone represented by a non-canonical DBM	150
5.43	A non-canonical result of computing $Z \swarrow$ in spite of the canonical input	151
5.44	Examples of borders of two zones.....	152
5.45	An algorithm for computing difference of zones.....	154
6.1	The model M considered in Example 6.1	157
6.2	The stages of labelling M with subformulas of $\varphi = E(\wp_1 U(EG\wp_2))$	157
6.3	A tree	159
6.4	The transition function of the automaton \mathcal{A} of Example 6.4....	161

6.5	A run $r = (\mathfrak{T}_r, V_r)$ of the automaton \mathcal{A} of Example 6.4 on a tree $(\mathfrak{T}, V_{\mathfrak{T}})$	162
6.6	The model M considered in Example 6.8	168
6.7	A part of the AND/OR graph of the automaton $\mathcal{A}_{M \times \varphi}$ for the model of Fig. 6.6 and $\varphi = E(\wp_1 U(EG\wp_2))$	169
6.8	Labelling the states while testing non-emptiness of the automaton $\mathcal{A}_{M \times \varphi}$ for the model of Fig. 6.6 and $\varphi = E(\wp_1 U(EG\wp_2))$	171
6.9	The timed automaton \mathcal{A}_{φ} augmenting that of Fig. 5.17 to verify TCTL formulas with one timing interval	173
6.10	The reachable part if the boundary-distinguishing detailed region graph for the automaton of Fig. 5.17 and $c_{max}(\mathcal{A}, \varphi) = 1$	175
6.11	Testing whether $q \models \gamma$	175
6.12	Testing γ translated to CTL^r_{-X}	176
7.1	Discretizing $[0, 1)^2$	184
7.2	Time- and action successor relation vs. discretization.	184
7.3	The timed automaton used in Example 7.3	186
7.4	The reachable part of the detailed region graph for the automaton of Fig. 7.3 and $c_{max}(\mathcal{A}, \varphi) = 1$	187
7.5	The sequence of transitions considered in Example 7.3 in the reachable part of the discretized model for the automaton in Fig. 7.3	187
7.6	Sequences of transitions in the reachable part of the discretized rg-model for the automaton in Fig. 7.3 and $c_{max}(\mathcal{A}, \varphi) = 1$	189
7.7	Sequences of transitions in the reachable part of the discretized rg _b -model for the automaton in Fig. 7.3 and $c_{max}(\mathcal{A}, \varphi) = 1$. . .	191
7.8	Elements of the k -model for $k = 2$	192
7.9	Two kinds of k -paths	192
7.10	Examples of $ECTL^r_{-X}$ formulas	193
7.11	Sets P'_k of submodels of M_k ($k = 2$)	194
7.12	An intuition behind the number of k -paths in $f_k(E(\alpha U_y \beta))$	195
7.13	An intuition behind the number of k -paths in $f_k(E(\alpha R_y \beta))$	195
7.14	An intuition behind the translations of TCTL formulas into $ECTL^r_{-X}$ formulas	199
7.15	An intuition behind the translation of $X_{z_i} \alpha$	199
7.16	A BMC algorithm for TECTL	200
7.17	Examples of $ECTL^r_{-X}$ formulas	200
7.18	A special k -path	201
7.19	A free special k -path	202
7.20	An intuition behind the translations of $\varphi = \alpha U \beta$ and $\varphi = \alpha R \beta$ when the k -path is not a loop	206
7.21	An intuition behind the translations of $\varphi = \alpha U \beta$ and $\varphi = \alpha R \beta$ when the k -path is a loop	206

7.22	An algorithm computing the characteristic state predicate for a $T\mu$ formula	210
7.23	An inequality graph for Example 7.16	215
7.24	The structure of a graph with exponentially many simple cycles	216
7.25	An accumulating chord	216
7.26	A generic SAT algorithm	221
7.27	An implication graph for φ	223
7.28	A generic equCNF algorithm	226
7.29	A dag for the formula $\varphi = \wp_1 \vee (\wp_2 \wedge \wp_3 \wedge (\wp_4 \Leftrightarrow \wp_5))$	227
7.30	Identifying the blocking clause $\wp_1 \vee \wp_2$	227
7.31	Identifying the blocking clause $\wp_1 \vee \wp_3$	228

List of Symbols

\cdot	concatenation, page 159
\rightarrow	successor relation of M , page 69
\rightarrow_{II}	successor relation of II , page 123
\rightarrow_{IIL}	successor relation between classes of II (Lee-Yannakakis algorithm, relates marked classes), page 132
\rightarrow_a	abstract transition relation, page 90
\rightarrow_c	concrete successor relation in $C(\mathcal{T})$, page 89
\rightarrow_c	transition relation of $C_c(\mathcal{A})$, page 38
\rightarrow_{d_1}	discrete successor relation of \mathcal{A} , page 40
\rightarrow_{d_2}	discrete successor relation of \mathcal{A} , page 40
$\rightarrow_{\mathfrak{d}}$	transition relation of $DM_{DRG}(\mathcal{A})$, page 188
$\rightarrow_{\mathfrak{d}, \mathcal{A}}$	part of $\rightarrow_{\mathfrak{d}}$ where transitions are labelled with elements of $A \cup \{\tau\}$, page 191
$\rightarrow_{\mathfrak{d}_c}$	transition relation of $DM(\mathcal{A})$, page 185
$\rightarrow_{\mathfrak{d}_y}$	part of $\rightarrow_{\mathfrak{d}}$ where transitions are labelled with a_y , page 191
\rightarrow_{F_c}	timed consecution relation in $C_c^F(\mathcal{N})$, page 16
\rightarrow_{N_c}	timed consecution relation in $C_c^N(\mathcal{N})$, page 14
\rightarrow_{P_c}	timed consecution relation in $C_c^P(\mathcal{N})$, page 13
\rightarrow_{T_c}	timed consecution relation of $C_c^T(\mathcal{N})$, page 11
$\rightarrow_{\triangleleft}$	successor relation in $DRG(\mathcal{A})$, page 118
$\rightarrow_{\triangleleft b}$	successor relation in $DRG_b(\mathcal{A})$, page 119
$\rightarrow_{\triangleleft b, \mathcal{A}}$	page 173
$\rightarrow_{\triangleleft b_{y_i}}$	page 173
$ \rightarrow $	the number of transitions of M , page 70
\equiv_G	equivalence of classes in geometric region graph, page 104
\equiv_S	equivalence of classes in SCG, page 98
\models	page 70
\models	satisfaction relation for $\mathcal{C}_{\mathcal{X}^+}^{\ominus}$, page 30
\models	satisfaction relation for $\mathcal{C}_{\mathcal{X}}^{\ominus}$, page 29
\sqsubset	ordering of bounds, page 146

$\simeq_{\mathcal{C}_{\mathcal{A},\varphi}}$	equivalence of clock valuations, page 116
$\lfloor \delta \rfloor$	integral part of δ , page 116
$\bar{\delta}$	negation of the variable δ , page 208
Δ	discretization step, page 183
Δ	transition function of \mathcal{A} , page 160
ϵ	label of adjust transitions in $DM(\mathcal{A})$, page 185
ϵ	empty sequence of transitions (state class methods), page 104
ϵ	empty sequence, page 159
η	sequence of transitions, element of C in some state class methods, page 103
η	node of a tree, page 159
ν	fictitious transition denoting start of \mathcal{N} , page 104
$\xi_{i,j}$	path from x_i to x_j in G_φ , page 216
π	path (of a tree), page 159
π	path, page 191
π	path in a CTL* model, page 70
π_i	page 70
$\pi(i)$	page 191
Π	partition of a state space, page 122
Π^{ps}	pseudoclasses of Π , page 127
Π_0	initial partition (minimization algorithms), page 122
$\Pi_k(s)$	set of all the k -paths starting at s , page 191
ρ	run of timed automaton, page 38
σ^F	concrete state in the firing interval semantics of \mathcal{N} , page 15
$(\sigma^F)^0$	initial state in $C_c^F(\mathcal{N})$, page 16
σ^N	concrete state in the semantics assigning clocks to the processes of \mathcal{N} , page 14
$(\sigma^N)^0$	initial state in $C_c^N(\mathcal{N})$, page 14
σ^P	concrete state in the semantics assigning clocks to the places of \mathcal{N} , page 12
$(\sigma^P)^0$	initial state in $C_c^P(\mathcal{N})$, page 13
σ^T	concrete state in the semantics assigning clocks to the transitions of \mathcal{N} , page 11
$(\sigma^T)^0$	initial state of $C_c^T(\mathcal{N})$, page 11
Σ^F	set of the concrete states in $C_c^F(\mathcal{N})$, page 16
Σ^N	set of the concrete states in $C_c^N(\mathcal{N})$, page 14
Σ^P	set of the concrete states in $C_c^P(\mathcal{N})$, page 13
Σ^T	set of the concrete states of $C_c^T(\mathcal{N})$, page 11
τ	label for time steps in time-abstracted models, page 94
$ \varphi $	the length (or size) of φ , page 69
$[\varphi]_k$	page 205
$[\varphi]_k^0$	page 205
$h[\varphi]_k^0$	page 205
$\llbracket \psi \rrbracket$	page 207
$\psi[\mathbf{A}]$	page 208

ψ_{bool}^a	page 211
ψ_{cons}^a	page 211
$[\psi]_{M_k}$	conjunct of $[M, \psi]_k$, page 195
$\psi_1 \leadsto \psi_2$	page 208
$\psi[a \leftarrow b]$	substitution (a is substituted with b in ψ)
a	element of \mathbb{A} , page 159
\underline{a}	page 203
a_{y_i}	label of \mathcal{A}_φ , page 172
\mathbf{A}	universal quantifier of CTL*, page 67
\mathcal{A}	set of the actions of \mathcal{A} , page 33
\mathbf{A}	assignment, page 219
\mathcal{A}	timed automaton, page 33
\mathcal{A}	alternating automaton over infinite trees, page 160
\mathbb{A}	labels (of a tree), finite alphabet, page 159
\mathcal{A}'	set of actions of \mathcal{A}_φ , page 172
\mathcal{A}_φ	\mathcal{A} extended to verify a TCTL formula φ , page 172
$\mathcal{A}_{\mathcal{N}}$	page 54, 56, 60
\mathbb{A}_r	labels of r , page 161
$\mathcal{A}_{\mathcal{D}, \varphi}$	page 164
$\mathcal{A}_{M \times \varphi}$	page 166
$A(a)$	indices of the components containing the action a , page 35
$\mathcal{A}_{i_1} \parallel \dots \parallel \mathcal{A}_{i_{n_j}}$	product of timed automata, page 35
\mathbf{AA}	alternating automata, page 158
ACTL	Universal Computation Tree Logic, page 68
ACTL*	universal CTL*, page 68
AE	condition AE, page 91
$AM_{\mathcal{N}}$	set of the markings of \mathcal{N} , page 8
$AM_{\mathcal{N}}^n$	set of the markings bounded by n of \mathcal{N} , page 8
$AM_{\mathcal{P}}$	set of the markings of \mathcal{P} , page 4
$AM_{\mathcal{P}}^m$	set of the markings of \mathcal{P} bounded by m , page 4
$b(q)$	set of the b -successors of q , page 123
$b^{-1}(q)$	set of the states for which q is b -successor, page 123
B	set of labels of successor relation \rightarrow_c in \mathcal{T} , page 89
$\mathcal{B}^+(Y)$	set of the positive boolean formulas over Y , page 159
$c_{max}(\mathcal{A}, \varphi)$	the largest constant in $\mathcal{C}_{\mathcal{A}}$ and time intervals in φ of TCTL or in clock constraints in φ of TCTL _C , page 116
$c_{max}(\mathcal{A})$	the largest constant appearing in $\mathcal{C}_{\mathcal{A}}$, page 116
\mathcal{C}	simple cycle, page 214
C	state class
C^0	initial state class
\mathcal{C}^\ominus	union of $\mathcal{C}_{\mathcal{X}}^\ominus$ and $\mathcal{C}_{\mathcal{X}^+}^\ominus$, page 30
$\mathcal{C}_{\mathcal{A}}$	clock constraints appearing in enabling conditions and invariants of \mathcal{A} , page 116
$\mathcal{C}_{\mathcal{X}}$	set of the clock constraints without clock differences (over \mathcal{X}), page 29

$\mathcal{C}_{\mathcal{X}}^{\ominus}$	set of the clock constraints over \mathcal{X} , page 29
$\mathcal{C}_{\mathcal{X}^+}^{\ominus}$	set of the normalised clock constraints, page 30
$\dot{C}(\mathcal{T})$	concrete state space of a system \mathcal{T} , page 89
$C_c(\mathcal{A})$	concrete dense state space of \mathcal{A} , page 38
$C_c^F(\mathcal{N})$	concrete state space in the firing interval semantics of \mathcal{N} , page 16
$C_c^N(\mathcal{N})$	concrete state space in the semantics assigning clocks to the processes of \mathcal{N} , page 14
$C_c^P(\mathcal{N})$	concrete state space in the semantics assigning clocks to the places of \mathcal{N} , page 13
$C_{d_r}^R(\mathcal{N})$	concrete (discrete) state space, page 22
$C_c^T(\mathcal{N})$	concrete state space in the semantics assigning clocks to the transitions of \mathcal{N} , page 11
cc	clock constraint, page 29
$\llbracket \mathbf{cc} \rrbracket$	set of the clock valuations satisfying cc , page 30
$cf(D)$	canonical form of DBM D , page 147
$clock^N$	function returning the time elapsed since the marked place of the process became marked most recently in \mathcal{N} , page 14
$clock^P$	function returning the time since a place became marked most recently in \mathcal{N} , page 12
$clock^T$	function returning the time elapsed since a transition became enabled most recently in \mathcal{N} , page 11
<i>Closure</i>	page 151
CNF	conjunctive normal form, page 218
$toCNF(\psi)$	page 219
$cor(w)$	core of the state w , page 91
$cr(\varphi)$	φ translated to CTL_{-X}^r , page 172
CTL	Computation Tree Logic, page 68
CTL*	Computation Tree Logic*, page 67
CTL ^z	page 177
CTL ^r _{-X}	modified CTL _{-X} , page 172
\mathbf{d}_{ij}	element of difference bound matrix, page 147
$d(\eta)$	degree of η , page 159
\mathcal{D}	set of degrees, page 159
\mathbb{D}	discretized clock space, page 183
$\llbracket D \rrbracket$	zone defined by DBM D , page 147
DBM	difference bound matrix, page 147
$DM(\mathcal{A})$	discretized (concrete) model for \mathcal{A} , page 185
$DM_{DRG}(\mathcal{A})$	discretized region graph model for \mathcal{A} , page 188
$DM_{DRG_b}(\mathcal{A})$	discretized boundary-distinguishing region graph model for \mathcal{A} , page 190
$dom(\mathbf{A})$	domain of \mathbf{A} , page 219
dpt	auxiliary function in minimization algorithm for simulating models, page 135
$dpt(w)$	depth of state w , page 92

DRG	detailed region graph, page 118
$DRG(\mathcal{A})$	detailed region graph for \mathcal{A} (weakly monotonic semantics), page 118
$DRG_b(\mathcal{A})$	boundary-distinguishing detailed region graph for \mathcal{A} (DRG for strongly monotonic semantics), page 119
$DZ(n_{\mathcal{X}})$	set of all the detailed zones for \mathcal{X} , page 117
\underline{e}	page 203
e_{δ}	page 203
e_{null}^i	page 203
\exists	existential quantifier of CTL*, page 67
\mathbb{E}	page 183
E	transition relation of \mathcal{A} , page 33
E'	transition relation of \mathcal{A}_{φ} , page 172
$E_{\mathcal{N}}$	page 54, 57
EA	condition EA, page 91
ECTL	Existential Computation Tree Logic, page 68
ECTL*	existential CTL*, page 68
EE	condition EE, page 90
EE ₁	condition EE ₁ , page 90
EE ₂	condition EE ₂ , page 90
Eft	the earliest firing time function of \mathcal{N} , page 8
$en(m)$	set of the transitions enabled at m in \mathcal{P} , page 5
$enc_{PL}(l)$	page 209
$extr(Z)$	extrapolation of zone Z , page 142
$extr_d(Z)$	extrapolation of zone Z (diagonal constraints case), page 144
$f_{\mathcal{A}}(q)$	set of the progressive (weakly monotonic) q -runs of \mathcal{A} , page 39
$f_{\mathcal{A}}^+(q)$	set of the (progressive) strongly monotonic q -runs of \mathcal{A} , page 39
f_k	function determining the number of k -paths of a submodel sufficient to check a formula, page 194
$f_{\mathcal{N}}(\sigma)$	set of the progressive weakly monotonic σ -runs of \mathcal{N} , page 20
$f_{\mathcal{N}}^+(\sigma)$	set of the progressive strongly monotonic σ -runs of \mathcal{N} , page 20
\mathbf{F}	eventually operator of CTL*, page 67
F	flow function of \mathcal{P} , page 3
F	acceptance condition for \mathcal{A} , page 160
fi	firing interval in \mathcal{N} , page 15
$Fill$	page 151
$frac(\delta)$	fractional part of δ , page 116
$free(l)$	condition to check deadlock-freedom of \mathcal{A} , page 45
FV	fixed-point variables of μ -calculus, page 71
G_{φ}	inequality graph for φ , page 213
\mathbf{G}	always operator of CTL*, page 67
$guard_{\mathcal{N}}$	page 54, 56, 60

$H(w, w')$	page 197
HAA	hesitant alternating automata, page 158
\mathcal{I}	location invariant of \mathcal{A} , page 33
I	set of inequalities, page 96
$\mathcal{I}_{\mathcal{A}}$	page 209
\mathfrak{I}_m	page 56, 57, 58
$\mathcal{I}_{\mathcal{N}}$	page 54, 56
$I_s(w)$	page 197
I_1, \dots, I_r	intervals in φ , page 172
$inv_{\mathcal{N}}$	page 56, 60
$Iqs(\mathcal{V})$	set of all the inequalities over \mathcal{V} , page 96
$IqSets(\mathcal{V})$	set of all the subsets of $Iqs(\mathcal{V})$, page 96
l^0	initial location of \mathcal{A} , page 33
l_{ψ}	page 219
l_b	length of a vector of global state variable, page 196
L	locations of \mathcal{A} , page 33
L_{μ}	modal μ -calculus, page 71
\mathfrak{L}_m	page 56
$L_{\mathcal{N}}$	page 54, 56
L_{-X}	logic L without the next-step operator X, page 68
$\mathcal{L}(\mathcal{A})$	language of \mathcal{A} , page 163
$L(k, h)$	page 205
$L_j(k, h)$	page 197
Lft	the latest firing time function of \mathcal{N} , page 8
$lit()$	page 196
$loop$	page 192
LTL	Linear-Time Temporal Logic, page 68
m	marking of \mathcal{P} , page 4
m^0	initial marking of \mathcal{P} , page 3
$m[t]$	t enabled at m in \mathcal{P} , page 5
M	CTL model, page 69
M_k	k -model, page 193
$[M]_k$	page 205
$ M $	size of M , page 70
$[M^{\psi, s^0}]_k$	conjunct of $[M, \psi]_k$, page 195
$[M, \psi]_k$	propositional formula to be tested to check ψ over M_k , page 195
$M_a(\mathcal{T})$	abstract model for \mathcal{T} , page 89
$M_c(\mathcal{A})$	concrete dense model for \mathcal{A} , page 183
$M_c(\mathcal{N})$	concrete model of \mathcal{N} , page 23
$M_c(\mathcal{T})$	concrete model of a given system \mathcal{T} , page 89
$M_{DRG_b}(\mathcal{A}_{\varphi})$	modified bd-region graph model for \mathcal{A}_{φ} , page 173
$M'_{DRG_b}(\mathcal{A}_{\varphi})$	extended $M_{DRG_b}(\mathcal{A}_{\varphi})$, page 174
\mathbb{N}	set of natural numbers, page 3
\mathfrak{N}	family of 1-safe sequential TPNs, page 9

\mathcal{N}	time Petri net, page 7
\mathbb{N}_+	set of positive natural numbers, page 3
\mathbb{N}^*	set of all the finite sequences of natural numbers, page 159
$newly_en(m, t)$	set of the transitions newly enabled after firing t , page 5
\mathcal{O}	page 161
ox	page 203
\wp_b	page 173
$\wp_{y_i \in I_i}$	new proposition for interval I_i , page 173
$\wp(w)$	page 197
\mathcal{P}	Petri net, page 3
P	set of places of \mathcal{P} , page 3
P_k	set of all the k -paths of a discretized $\text{rg}(b)$ model, page 192
P_y	set of all the pairs of states $(s, s[y := 0])$ of a discretized $\text{rg}(b)$ model, page 192
pAE	condition pAE, page 92
$parent(v_t, C)$	variable for the parent of t in C , page 104
PL	propositional logic, page 63
$Post_{\Pi}(Y)$	successors of Y in Π , page 123
$post_b(Y, Y')$	page 123
$Pre_{\Pi}(Y)$	predecessors of Y in Π , page 123
$pre_{\mathcal{A}}(\psi)$	page 209
$pre_b(Y, Y')$	page 123
$pre_e(\psi)$	page 209
pU	condition pU, page 92
PV	propositional variables of PL, page 64
PV'	extended PV, page 173
PV_{φ}	set of the new propositions for intervals in φ , page 173
PV_P	set of the propositional variables corresponding to P of \mathcal{N} , page 23
$PV(\varphi)$	set of propositions occurring in φ , page 67
q^0	initial state of \mathcal{A} , page 160
q^0	initial state of $C_c(\mathcal{A})$, page 38
\mathcal{Q}	states of \mathcal{A} , page 160
Q	set of the concrete states of $C_c(\mathcal{A})$, page 38
\mathbb{Q}_+	set of positive rational numbers, page 3
\mathbb{Q}_{0+}	set of non-negative rational numbers, page 3
QPL	quantified propositional logic, page 65
QSL	quantified separation logic, page 66
r	run of \mathcal{A} , page 161
r	number of the intervals in φ , page 172
r_Y	representative of the class Y of Π , page 130
R	release operator of CTL*, page 67
\mathcal{R}	bisimulation of concrete state spaces of TPNs, page 17
\mathbb{R}	set of rational numbers, page 3
\mathbb{R}	set of real numbers, page 3

R	region, page 123
\mathbb{R}_+	set of positive real numbers, page 3
\mathbb{R}_{0+}	set of non-negative real numbers, page 3
$R(n_{\mathcal{X}}, L)$	set of the regions over \mathcal{X} and L , page 123
$\mathcal{R}(w, w')$	page 197
$R_x(w, w')$	page 197
$R \setminus R'$	difference of regions R, R' , page 123
$Reach_{\mathcal{A}}$	set of the reachable states of \mathcal{A} , page 39
$Reach_{\mathcal{A}}^+$	set of the reachable states on the strongly monotonic q^0 -runs of \mathcal{A} , page 39
$Reach_{\mathcal{A}}^{dr}$	reachable states on the discrete runs of \mathcal{A} , page 40
$Reach_{\mathcal{N}}^R$	set of the reachable states of \mathcal{N} , page 19
$Reach_{\mathcal{N}}^{+R}$	set of the states reachable on the strongly monotonic runs of \mathcal{N} ($R \in \{F, P, N, T\}$ and corresponds to the semantics considered), page 20
$Reach_{\mathcal{N}}^{Rdr}$	set of the reachable states on the discrete runs of \mathcal{N} , page 21
$Reach(\mathcal{T})$	set of all the states reachable in a given concrete state space $C(\mathcal{T})$ for \mathcal{T} , page 89
$Reach(\mathcal{U})$	set of all the reachable elements of the set family \mathcal{U} , page 89
$reset_{\mathcal{N}}$	page 54, 57, 60
$restrict$	page 153
$Rlx(C)$	page 113
$RM_{\mathcal{N}}$	set of the reachable markings of \mathcal{N} , page 19
$RM_{\mathcal{N}}^+$	set of the reachable markings on the strongly monotonic runs of \mathcal{N} , page 20
$RM_{\mathcal{P}}^{k_{\mathcal{P}}}$	set of all the reachable markings of \mathcal{P} for the capacity $k_{\mathcal{P}}$, page 5
s^0	initial state of M , page 69
$s[y := 0]$	page 191
S	set of states of M , page 69
$S_{\mathbb{D}}$	set of the states of $DM(\mathcal{A})$, page 185
$ S $	the number of states of M , page 70
SC	specification clock variables, page 80
$SF(\varphi)$	set of the subformulas of φ , page 69
SL	separation logic or difference logic, page 65
$sol(I)$	set of the solutions of I , page 96
Sp_s	auxiliary function to split classes in minimization algorithm for simulating models, page 133
$Succ_a$	page 138
$succ(s)$	page 166
$t \bullet$	postset of transition t , page 4
$\bullet t$	preset of transition t , page 4
\mathfrak{T}	tree, page 159
\mathcal{T}	timed system, page 89
T	set of transitions of \mathcal{P} , page 3

T^*	set of all the finite sequences of transitions of T , page 103
\mathfrak{T}_r	tree of a run r , page 161
$T\mu$	timed μ -calculus, page 80
TCTL	timed CTL, page 75
TCTL _C	alternative timed CTL, page 82
U	Until operator of CTL*, page 67
U	condition U, page 91
\mathbb{U}^{n_x}	valuations to represent detailed zones in discretized model, page 183
v	clock valuation, page 29
v_t	variable corresponding to transition t (state class approaches), page 97
v_t^j	variable for j -th firing of t (state class methods)
$v(i)$	value of clock x_i in v , page 29
$v(x_i)$	value of clock x_i in v , page 29
$v[X := 0]$	valuation like v where the clocks of X are set to 0, page 29
V	valuation function of PL, page 64
V_φ	function labelling states of $M_{DRG_b}(\mathcal{A}_\varphi)$ with propositions from PV_φ and with \wp_b , page 173
V_a	valuation function in $M_a(T)$, page 90
$V_{\mathcal{A}}$	valuation function for \mathcal{A} , page 41
V_c	valuation function for a concrete state space of \mathcal{A} , page 41
$V_{\mathcal{C}}$	valuation function in $M_c(T)$, page 89
$V_{\mathfrak{D}}$	valuation function of $DM_{DRG}(\mathcal{A})$, page 188
$V_{\mathfrak{D}_c}$	valuation function in $DM(\mathcal{A})$, page 185
$V_{\mathcal{N}}$	valuation function for \mathcal{N} , page 23
V_r	valuation function of \mathfrak{T}_r , page 161
$V_{\mathfrak{T}}$	valuation function (a tree), page 159
V'_a	extended V_a , page 173
$V_{\mathcal{A}_{\mathcal{N}}}$	page 54, 57, 61
(V, v)	model for (Q)SL, page 66
$V[\wp \leftarrow b]$	substitution in QPL, page 65
$var(I)$	set of the variables appearing in I , page 96
w	abstract state
w	global state variable (vector encoding a state), page 196
w^0	initial state of abstract model, page 90
$w[i]$	state variable, page 196
W	states of abstract model, page 90
WAA	weak alternating automata, page 158
x_0	fictitious clock representing constant 0, page 30
\mathcal{X}	set of clocks, page 29
X	next-step operator of CTL*, page 67
\mathcal{X}'	set of clocks of \mathcal{A}_φ , page 172
\mathcal{X}^+	set of clocks together with x_0 , page 30
R_{y_i}	release operator of CTL*_X, page 174

XXXII List of Symbols

U_{y_i}	until operator of CTL^r_{-X} , page 174
X_{z_i}	next-step operator of CTL^z , page 177
$\mathcal{X}_{\mathcal{N}}$	page 54, 56, 60
$[X := 0]Z$	page 32
y_i	auxiliary clock in \mathcal{A}_{φ} , page 172
Y	class of Π , page 122
z	absolute time reference, page 203
\mathbb{Z}	set of integers, page 3
Z^0	initial detailed zone, page 118
Z_0	page 138
\mathbb{Z}_+	set of positive integers, page 3
$Z \nearrow$	page 31
$Z \swarrow$	page 32
\mathbb{Z}_{0+}	set of non-negative integers, page 3
$Z(n_{\mathcal{X}})$	zones for \mathcal{X} , page 31
$Z \uparrow Z'$	page 32
$Z[X := 0]$	page 32