

Guanjun Liu

Petri Nets

Theoretical Models and Analysis
Methods for Concurrent Systems

 Springer

Petri Nets

Guanjun Liu

Petri Nets

Theoretical Models and Analysis Methods for
Concurrent Systems

 Springer

Guanjun Liu 
Department of Computer Science
Tongji University
Shanghai, China

ISBN 978-981-19-6308-7 ISBN 978-981-19-6309-4 (eBook)
<https://doi.org/10.1007/978-981-19-6309-4>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, 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.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Preface

Petri nets, as the first true concurrent model, have been of good vitality since Dr. Carl Adam Petri pioneered them sixty years ago. Such a good vitality benefits from the outstanding work of Dr. Petri and other scientists and technicians in this field, indicating that Petri nets (including high-level extensions) have many advantages in modelling concurrent systems as well as in helping us understand, analyse, and improve these concurrent systems.

Concurrency is a universal phenomenon in our physical world. Even when an organisation or a team performs a task, concurrent operations are often taken in order to increase efficiency. In the current scientific times, lots of concurrent systems serve our work and life, but they must provide a correct and reliable service, or our safety and even lives are threatened. However, designing such a system is not an easy thing, because a number of concurrent, interactive, and even real-time actions are involved in its running so that its logical structures and dynamic behaviours are very complex. In consequence, we need one or multiple suitable models to help us understand and analyse the logical structures and dynamic behaviours of such a complex system, while Petri nets have been well used just due to their ability on simulating concurrence, interaction, real time, etc.

Purpose

In addition to a part of my research results, what this book focuses on is the basic theory of Petri nets as well as the Petri-nets-based model checking methods, so that it is suitable not only for senior researchers, but also for beginners aspiring to this field. The biggest feature of this book, I think, is a relatively systematic combination of the Petri net theory and the Petri-nets-based model checking methods. The Petri net theory can help readers systematically understand the nature of concurrence, interaction, and real time. The model checking methods are important practical techniques employed for checking errors in concurrent systems. So far, only few books combine them together.

Prerequisites

The Petri net theory and model checking methods have their roots in mathematical foundations such as discrete mathematics, propositional logic, automata theory, data structures, graph theory, and algorithms. It is expected that readers are familiar with the basic knowledge of these aspects when starting with this book.

Content

As for the Petri net theory, this book involves the basic knowledge of elementary net systems, including their interleaving semantics and concurrency semantics (Chap. 1), some structure concepts (Chap. 2), some subclasses with special structures (Chaps. 3 and 4), and some basic properties such as reachability, liveness, and deadlock (Chaps. 1–4). It also involves four high-level Petri nets: Knowledge-oriented Petri nets (Chap. 6), Petri nets with insecure places (Chap. 7), time Petri nets (Chap. 8), and plain time Petri nets with priorities (Chap. 10), focusing on different application fields. As for the model checking methods, this book involves computation tree logic (Chap. 5), computation tree logic of knowledge (Chap. 6), timed computation tree logic (Chaps. 9 and 10), as well as the Petri-nets-based checking methods for them. The basic principle of the reduced ordered binary decision diagram is also introduced in order to compress the state space used in these model checking procedures (Chaps. 5 and 6). Besides the above contents, this book also presents time-soundness for time Petri nets (Chap. 8) and secure bisimulation for Petri nets with insecure places (Chap. 7). They are both based on the bisimulation theory pioneered by Dr. Robin Milner, and thus this theory is also introduced briefly in this book.

Acknowledgements

I would like to thank all my colleagues for their hard work in fighting against COVID-19. COVID-19 forced me to stay at home; but on the other hand, I had enough time to write this book. I thank the National Nature Science Foundation of China (Nos. 62172299 and 62032019) and the Alexander von Humboldt Foundation of Germany. They supported my research so that a part of my research results can be written in this book. At last, I thank my wife, my daughters, and my mother for their selfless support.

Shanghai, China
July 2022

Guanjun Liu

Contents

1	Elementary Net Systems	1
1.1	Net Diagram and Semantics	1
1.1.1	Net and Net System	1
1.1.2	Firable Transition Sequence and Interleaving Semantics	4
1.1.3	Process and Concurrency Semantics	8
1.2	Reachability Graph and Coverability Graph	14
1.2.1	Reachability and Reachability Graph	14
1.2.2	Unboundedness and Coverability Graph	16
1.3	Unfolding	18
1.3.1	Branching Process	18
1.3.2	Finite Complete Prefix	24
1.3.3	Finite Prefix	26
1.4	Basic Properties and Computation Complexity	27
1.4.1	Some Basic Properties	28
1.4.2	Computation Complexity of Deciding Basic Properties	29
1.5	Application	30
1.6	Summary and Further Reading	31
	References	32
2	Structural Characteristics of Petri Nets	35
2.1	Incidence Matrix and State Equation	35
2.1.1	Incidence Matrix	35
2.1.2	State Equation	37
2.2	Invariant	39
2.2.1	T-Invariant	40
2.2.2	P-Invariant	42
2.2.3	Algorithm of Computing Invariants	43

2.3	Repetitiveness	46
2.3.1	Repetitive Vector	46
2.3.2	Relation Between Repetitive Vector and T-Invariant	48
2.3.3	Algorithm of Computing Repetitive Vectors	50
2.4	Siphon and Trap	53
2.4.1	Siphon	53
2.4.2	Trap	54
2.4.3	Relation Between Siphon and Repetitive Vector	55
2.4.4	Algorithm of Computing Siphon and Trap	60
2.5	Application	62
2.6	Summary and Further Reading	66
	References	67
3	Petri Nets with Special Structures	69
3.1	Single Input and Single Output	69
3.1.1	State Machine	69
3.1.2	Marked Graph	72
3.2	Choice Structures	76
3.2.1	Free-Choice Net	76
3.2.2	Asymmetric-Choice Net	80
3.3	Circuit Structure	82
3.3.1	Normal Net	83
3.3.2	Weakly Persistent Net	88
3.4	Application	93
3.5	Summary and Further Reading	95
	References	96
4	Petri Nets Modeling Message Passing and Resource Sharing	99
4.1	Workflow Nets	99
4.1.1	Soundness and Weak Soundness of Workflow Nets	99
4.1.2	Equivalence Between Soundness and Weak Soundness in Free-choice Workflow Nets	102
4.1.3	Equivalence Between Soundness and Weak Soundness in Acyclic Asymmetric-Choice Workflow Nets	103
4.1.4	k -Soundness	104
4.2	Inter-organisational Workflow Nets	106
4.2.1	Compatibility	106
4.2.2	Collaborative-ness	108
4.3	Resource Allocation Nets	110
4.4	Application	113
4.5	Summary and Further Reading	118
	References	119

- 5 Verifying Computation Tree Logic Based on Petri Nets** 123
 - 5.1 Computation Tree Logic and Verification Based on Petri Nets 123
 - 5.1.1 Syntax and Semantics of CTL 123
 - 5.1.2 Logical Equivalence of Formulae 128
 - 5.1.3 Verification Algorithms Based on Reachability Graphs 130
 - 5.2 Reduced Ordered Binary Decision Diagrams 134
 - 5.2.1 Reduced Ordered Binary Decision Diagrams for Encoding Boolean Functions 134
 - 5.2.2 Constructing the ROBDD of a Boolean Function 138
 - 5.2.3 Operations of Boolean Functions by Manipulating ROBDDs 144
 - 5.3 Verifying CTL with the ROBDD Technique 150
 - 5.3.1 Encoding Reachability Graphs of Safe Petri Nets Using ROBDDs 150
 - 5.3.2 CTL Verification Based on All Reachable Markings Encoded by an ROBDD 159
 - 5.4 Application 163
 - 5.5 Summery and Further Reading 165
 - References 165

- 6 Knowledge-Oriented Petri Nets and Computation Tree Logic of Knowledge** 169
 - 6.1 Knowledge-Oriented Petri Nets Modelling Privacy-Critical Multi-agent Systems 169
 - 6.1.1 Knowledge-Oriented Petri Nets 169
 - 6.1.2 Reachability Graph with Equivalence Relations 172
 - 6.2 Computation Tree Logic of Knowledge 175
 - 6.2.1 Syntax and Semantics of CTLK 175
 - 6.2.2 Verifying CTLK Based on Reachability Graph with Equivalence Relations 180
 - 6.2.3 Verifying CTLK Based on ROBDD 183
 - 6.3 Applications 186
 - 6.3.1 Key Transport Protocol with Attacking 186
 - 6.3.2 Dining Cryptographers Protocol 187
 - 6.4 Summery and Further Reading 189
 - References 189

- 7 Petri Nets with Insecure Places and Secure Bisimulation** 191
 - 7.1 Bisimulation and Weak Bisimulation 191
 - 7.2 Petri Nets with Insecure Places and Secure Bisimulation 194
 - 7.2.1 Petri Nets with Insecure Places 194
 - 7.2.2 Secure Bisimulation 197
 - 7.3 Summery and Further Reading 201
 - References 201

8	Time Petri Nets and Time-Soundness	203
8.1	Time Petri Nets	203
8.1.1	Formal Definition and Firing Rules	203
8.1.2	Generation of State Class Graph	208
8.1.3	Completeness of State Class Graph	215
8.2	Time-Soundness	217
8.2.1	Time-Soundness Based on Bisimulation	217
8.2.2	Deciding Time-Soundness	224
8.3	Application	226
8.4	Discussion of CTL over Time Petri Nets	232
8.5	Summery and Further Reading	234
	References	235
9	Verifying Timed Computation Tree Logic Based on Time Petri Nets	237
9.1	Syntax and Semantics of TCTL	237
9.1.1	Syntax of TCTL	237
9.1.2	Semantics of TCTL	239
9.2	Verifying TCTL Based on State Class Graphs of Time Petri Nets	240
9.2.1	Recursively Computing Satisfiable State Sets of TCTL Formulae	240
9.2.2	Computing Satisfiable State Set for EU_1 Based on Forward Exploration	244
9.2.3	Computing Satisfiable State Set of AU_1 Based on Forward Exploration	253
9.3	Summery and Further Reading	256
	References	256
10	Plain Time Petri Nets with Priorities	259
10.1	Plain Time Petri Net with Priorities	259
10.1.1	Definition of Plain Time Petri Net with Priorities	259
10.1.2	Firing Rules and State Class Graph of Plain Time Petri Nets with Priorities	261
10.2	Properties of State Class Graphs of Plain Time Petri Nets with Priorities	266
10.3	Verifying TCTL Based on State Class Graphs of Plain Time Petri Nets with Priorities	267
10.3.1	Computing Satisfiable State Set for EU_1 Based on Backward Exploration	268
10.3.2	Computing Satisfiable State Set for AU_1 Based on Backward Exploration	270

10.4 Application	271
10.5 Summery and Further Reading	275
References	275
Index	277