Systems and Software Verification

Springer-Verlag Berlin Heidelberg GmbH

B. Bérard · M. Bidoit · A. Finkel · F. Laroussinie
A. Petit · L. Petrucci · Ph. Schnoebelen
with P. McKenzie

# Systems and Software Verification

## Model-Checking Techniques and Tools

With 67 Figures

Springer

Béatrice Bérard
Michel Bidoit
Alain Finkel
François Laroussinie
Antoine Petit
Laure Petrucci
Philippe Schnoebelen

Laboratoire Spécification et Vérification
CNRS, UMR 8643
Ecole Normale Supérieure de Cachan
61, avenue du Président Wilson
94235 Cachan Cedex, France
http://www.lsv.ens-cachan.fr/

Pierre McKenzie

Département d'Informatique
et Recherche Opérationnelle
Université de Montréal
CP 6128 succ Centre-Ville
Montréal QC H3C 3J7, Canada
http://www.iro.umontreal.ca/~mckenzie/

Translated with the help of Pierre McKenzie, Université de Montréal

Updated version of the French language edition:
"Vérification de logiciels. Techniques et outils du model-checking",
coordonné par Philippe Schnoebelen
Copyright © Vuibert, Paris, 1999
Tous droits réservés

Library of Congress Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

Systems and software verification: model-checking techniques and tools / Bérard ...

ISBN 978-3-642-07478-3 ISBN 978-3-662-04558-9 (eBook)
DOI 10.1007/978-3-662-04558-9

ACM Computing Classification (1998): D.2.4, D.2, D.4.5, F.3.1–2, F.4.1, G.4, I.2.2

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 for prosecution under the German Copyright Law.

http://www.springer.de

© Springer-Verlag Berlin Heidelberg 2001
Originally published by Springer-Verlag Berlin Heidelberg New York in 2001
Softcover reprint of the hardcover 1st edition 2001

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: KünkelLopka, Heidelberg
Typesetting: Camera-ready by authors using a Springer TEX macro package
Printed on acid-free paper SPIN 11395898 41/3111/XO – 5 4 3 2 1

# Foreword

One testament to the maturing of a research field is the adoption of its techniques in industrial practice; another is the emergence of textbooks. According to both signs, research in model checking is now entering its mature phase, some twenty years after almost entirely theoretical beginnings. It has been an exciting twenty years, which have seen the research focus evolve, like the business plan of a successful enterprise, from a dream of automatic program verification to a reality of computer-aided design debugging.

Those who have participated in significant hardware designs or embedded software projects have experienced that the system complexity, and hence the likely number of design errors, grows exponentially with the number of interacting system components. Furthermore, traditional debugging and validation techniques, based on simulation and testing, are woefully inadequate for detecting errors in highly concurrent designs. It is therefore in such applications that model-checking-based techniques, despite their limitations in the face of the exponential complexity growth, are making inroads into the design flow and the supporting software tools.

This monograph, with its emphasis on skills, craft, and tools, will be of particular value, first, to the practitioner of formal verification, and second, as a textbook for courses on formal verification that —as, I believe, all courses on formal verification should— contain a significant experimental component. The selected model-checking tools are available to the public, free of charge, and are suitable for use in the classroom. The reader is exposed to a broad mix of different tools, from widely used, mature software to experimental tools at the research frontier, including programs for model checking real-time and hybrid systems. In this way, the book succeeds in providing both: a survey of established techniques in model checking, as well as a glimpse at state-of-the-art research.

Berkeley, February 2001 *Thomas A. Henzinger*

# Preface

This book is an introduction to *model checking*, a technique for automatic verification of software and reactive systems. Model checking was invented more than twenty years ago. It was first developed by academic research teams and has more recently been introduced in specialized industrial units. It has now proven to be a successful method, frequently used to uncover well-hidden bugs in sizeable industrial cases. Numerous studies are still in progress, both to extend the area covered by this technique and to increase its efficiency. This leads us to believe that its industrial applications will grow significantly in the next few years.

The book contains the basic elements required for understanding model checking and is intended both as a textbook for undergraduate courses in computer science and as a reference for professional engineers. It may also be of interest to researchers, lecturers or PhD students wishing to prepare a talk or an overview on the subject. To increase his theoretical knowledge on model checking, the reader is invited to consult the excellent monograph *Model Checking* (MIT Press), by Clarke, Grumberg and Peled, which had not yet been published when the French edition of this book appeared.

The first part of the book presents the fundamental principles underlying model checking. The second part considers the problem of specification in more detail. It provides some help for the design of temporal logic formulae to express classes of properties, which are widely used in practice. The third part describes, from a user's point of view, some significant model checkers freely available in the academic world. All of them have been used by the authors in the course of their industrial collaborations.

This book was written by French researchers from *Laboratoire Spécification et Vérification* (LSV), a joint laboratory of École Normale Supérieure de Cachan and Centre National de la Recherche Scientifique. It is a revised translation of *Vérification de logiciels : Techniques et outils du model-checking*, (Vuibert, 1999), a former French undergraduate/graduate textbook co-ordinated by Philippe Schnoebelen and written by Béatrice Bérard, Michel Bidoit, François Laroussinie, Antoine Petit and Philippe Schnoebelen, with the help of Gérard Cécé, Catherine Dufourd, Alain Finkel, Laure Petrucci and Grégoire Sutre. The French book was itself derived from a collaboration between the French Electricity Company (EDF) and the LSV.

*Acknowledgements.* Gérard Cécé, Roopa Chauhan, Sandrine Couffin, Catherine Dufourd, Paul Gastin, Jean Goubault-Larrecq, Jean-Michel Hufflen, Amélie Josselin, Pierre McKenzie, Christine Pellen, Sophie Pinchinat, Marie-Pierre Ponpon, Jérôme Ryckbosch and Grégoire Sutre read the first versions of the book. Their numerous remarks and suggestions were invaluable to the production of the final version.

# Contents

## Part I. Principles and Techniques


**Introduction** ..... 3

**1. Automata** ..... 5
- 1.1 Introductory Examples ..... 5
- 1.2 A Few Definitions ..... 9
- 1.3 A Printer Manager ..... 11
- 1.4 A Few More Variables ..... 12
- 1.5 Synchronized Product ..... 14
- 1.6 Synchronization by Message Passing ..... 21
- 1.7 Synchronization by Shared Variables ..... 24

**2. Temporal Logic** ..... 27
- 2.1 The Language of Temporal Logic ..... 28
- 2.2 The Formal Syntax of Temporal Logic ..... 32
- 2.3 The Semantics of Temporal Logic ..... 33
- 2.4 PLTL and CTL: Two Temporal Logics ..... 35
- 2.5 The Expressivity of CTL* ..... 37

**3. Model Checking** ..... 39
- 3.1 Model Checking CTL ..... 39
- 3.2 Model Checking PLTL ..... 42
- 3.3 The State Explosion Problem ..... 45

**4. Symbolic Model Checking** ..... 47
- 4.1 Symbolic Computation of State Sets ..... 47
- 4.2 Binary Decision Diagrams (BDD) ..... 51
- 4.3 Representing Automata by BDDs ..... 54
- 4.4 BDD-based Model Checking ..... 56

**5. Timed Automata** ..... 59
- 5.1 Description of a Timed Automaton ..... 60
- 5.2 Networks of Timed Automata and Synchronization ..... 62

5.3 Variants and Extensions of the Basic Model ..... 64
5.4 Timed Temporal Logic ..... 67
5.5 Timed Model Checking ..... 68

**Conclusion** ..... 73

**Part II. Specifying with Temporal Logic**

**Introduction** ..... 77

**6. Reachability Properties** ..... 79
6.1 Reachability in Temporal Logic ..... 79
6.2 Model Checkers and Reachability ..... 80
6.3 Computation of the Reachability Graph ..... 80

**7. Safety Properties** ..... 83
7.1 Safety Properties in Temporal Logic ..... 83
7.2 A Formal Definition ..... 84
7.3 Safety Properties in Practice ..... 86
7.4 The History Variables Method ..... 87

**8. Liveness Properties** ..... 91
8.1 Simple Liveness in Temporal Logic ..... 92
8.2 Are Liveness Properties Useful? ..... 92
8.3 Liveness in the Model, Liveness in the Properties ..... 94
8.4 Verification under Liveness Hypotheses ..... 96
8.5 Bounded Liveness ..... 97

**9. Deadlock-freeness** ..... 99
9.1 Safety? Liveness? ..... 99
9.2 Deadlock-freeness for a Given Automaton ..... 99
9.3 Beware of Abstractions! ..... 101

**10. Fairness Properties** ..... 103
10.1 Fairness in Temporal Logic ..... 103
10.2 Fairness and Nondeterminism ..... 104
10.3 Fairness Properties and Fairness Hypotheses ..... 104
10.4 Strong Fairness and Weak Fairness ..... 106
10.5 Fairness in the Model or in the Property? ..... 107

**11. Abstraction Methods** ..... 109
11.1 When Is Model Abstraction Required? ..... 110
11.2 Abstraction by State Merging ..... 110
11.3 What Can Be Proved in the Abstract Automaton? ..... 110
11.4 Abstraction on the Variables ..... 114

11.5 Abstraction by Restriction ..... 118
11.6 Observer Automata ..... 120

**Conclusion** ..... 125

**Part III. Some Tools**

**Introduction** ..... 129

**12. SMV – Symbolic Model Checking** ..... 131
12.1 What Can We Do with SMV? ..... 131
12.2 SMV's Essentials ..... 131
12.3 Describing Automata ..... 132
12.4 Verification ..... 135
12.5 Synchronizing Automata ..... 136
12.6 Documentation and Case Studies ..... 137
SMV Bibliography ..... 138

**13. SPIN – Communicating Automata** ..... 139
13.1 What Can We Do with SPIN? ..... 139
13.2 SPIN's Essentials ..... 139
13.3 Describing Processes ..... 140
13.4 Simulating the System ..... 141
13.5 Verification ..... 142
13.6 Documentation and Case Studies ..... 144
SPIN Bibliography ..... 144

**14. DESIGN/CPN – Coloured Petri Nets** ..... 145
14.1 What Can We Do with DESIGN/CPN? ..... 145
14.2 DESIGN/CPN's Essentials ..... 145
14.3 Editing with DESIGN/CPN ..... 146
14.4 Simulating the Net ..... 147
14.5 Analyzing the Net ..... 149
14.6 Documentation and Case Studies ..... 149
DESIGN/CPN Bibliography ..... 150

**15. UPPAAL – Timed Systems** ..... 153
15.1 What Can We Do with UPPAAL? ..... 153
15.2 UPPAAL's Essentials ..... 153
15.3 Modeling Timed Systems with UPPAAL ..... 154
15.4 Simulating a System ..... 157
15.5 Verification ..... 157
15.6 Documentation and Case Studies ..... 158
UPPAAL Bibliography ..... 158

**16. KRONOS – Model Checking of Real-time Systems** ..... 161
16.1 What Can We Do with KRONOS? ..... 161
16.2 KRONOS' Essentials ..... 161
16.3 Describing Automata ..... 162
16.4 Synchronized Product ..... 164
16.5 Model Checking ..... 165
16.6 Documentation and Case Studies ..... 167
KRONOS Bibliography ..... 167

**17. HYTECH – Linear Hybrid Systems** ..... 169
17.1 What Can We Do With HYTECH? ..... 169
17.2 HYTECH's Essentials ..... 169
17.3 Describing Automata ..... 170
17.4 System Analysis ..... 172
17.5 Parametric Analysis ..... 174
17.6 Documentation and Case Studies ..... 176
HYTECH Bibliography ..... 176

**Main Bibliography** ..... 179

**Index** ..... 183