LNCIS

LECTURE NOTES IN CONTROL
AND INFORMATION SCIENCES

356

Andrei Karatkevich

Dynamic Analysis of Petri Net-Based Discrete Systems

Springer

Andrei Karatkevich

Dynamic Analysis of Petri Net-Based Discrete Systems

Springer

Author

Dr. Andrei Karatkevich

Institute of Computer Engineering and Electronics
Faculty of Electrical Engineering,
Computer Science and Telecommunications
University of Zielona Góra
Ul. Pogfórna 50
65-246 Zielona Góra
Poland
Email: A.Karatkevich@IIE.UZ.ZGORA.PL

Library of Congress Control Number: 2007923719

ISSN print edition: 0170-8643
ISSN electronic edition: 1610-7411
ISBN-10 3-540-71464-2 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-71464-4 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Typesetting: by the authors and SPS using a Springer LaTeX macro package

Printed on acid-free paper SPIN: 11903901 89/SPS 5 4 3 2 1 0

Lecture Notes in Control and Information Sciences 356

Preface

Design of modern digital hardware systems and of complex software systems is almost always connected with parallelism. For example, execution of an object-oriented program can be considered as parallel functioning of the co-operating objects; all modern operating systems are multitasking, and the software tends to be multithread; many complex calculation tasks are solved in distributed way. But designers of the control systems probably have to face parallelism in more evident and direct way. Controllers rarely deal with just one controlled object. Usually a system of several objects is to be controlled, and then the control algorithm naturally turns to be parallel.

So, classical and very deeply investigated model of discrete device, Finite State Machine, is not expressive enough for the design of control devices and systems. Theoretically in most of cases behavior of a controller can be described by an FSM, but usually it is not convenient; such FSM description would be much more complex, than a parallel specification (even as a network of several communicating FSMs).

The engineers and researchers became aware of practical necessity of developing of parallel discrete models about forty years ago. There were (and are) two main approaches to such models. One is a direct development of the FSM, being enhanced by parallelism and hierarchy. Another one is based on the parallelism "from the very beginning". The most famous and popular model of this second kind is Petri nets. A big family of more or less detailed behavioral specifications of parallel systems is based on this formalism.

Design and, especially, analysis and verification of systems, which behavior is specified by the parallel models, is a remarkably more complex task, than design and analysis of strictly sequential systems.

In this book, we are concerned about the formal analysis and verification of the parallel systems, specified by the Petri nets and the extended Petri net models. Besides, some results presented here are related to the FSM networks (but, again, we model them by means of Petri nets) and to the sequent automata (a kind of parallel descriptions other than Petri nets). To formulate some general affirmations, we use a general model of a parallel discrete system, which covers all specific models studied in the book.

We have focused on the approach of reduced exploration of state spaces. This approach is selected here as the basic one, because the state exploration provides the most detailed information about system behavior among other analysis approaches, and, on the other hand, such exploration does not have to be full to decide many important properties of the systems (especially with restricted structure).

The analysis methods of such kind are thoroughly developed; our work was inspired by the results of many authors, first of all of A. Valmari and P. Godefroid. For inspiration of another kind (the interesting parallel models and the methodology of research) we are grateful to A. Zakrevskij. The original results presented in this book are mostly connected to generalization of the known methods or, vice versa, to applying them to specific subclasses of parallel systems, which sometimes allows to obtain more information than in general case.

We intended this book to be useful to CAD researches and designers of parallel control systems. Content of the book is mostly theoretical, but it was written bearing in the mind possible practical applications. It may also be useful for the students of electrical engineering and computer science.

March 2007

Andrei Karatkevich
Zielona Góra

Symbols

Main symbols

A	FSM or parallel automaton
C	incidence matrix of a Petri net
D	siphon
e	event
E	set of edges of a graph
F	set of arcs of a Petri net
G	graph
I	set of input (external) events
k	elementary conjunction; implicant of a Boolean function
L	cycle in a graph
M	marking or global state
M_d	deadlock
M_0	initial marking or initial global state
mp	macroplace
N	FSM network
O	set of output events
p	place or local state
$\mathbf{p}_i$	Boolean variable corresponding to place or local state p_i
P	set of places or set of local states
P^{in}	set of input places
P^{out}	set of output places
Q	path in a graph
Qp_i	code of local state p_i (an elementary conjunction)
R	partial order relation
s	sequent
S	sequent automaton
t	transition
T	set of transitions
T_P	persistent set
T_S	stubborn set

u	elementary disjunction
V	set of nodes of a graph
w	weight (of an arc)
X	set of input variables
Y	set of output variables
Z	set of internal variables or events
$\mathit{\Gamma}$	set of events
$\mathit{\Delta}$	step of concurrent simulation
ϵ_i	Boolean variable corresponding to event e_i
μ	initial label of a parallel automaton transition
ν	terminal label of a parallel automaton transition
ρ	priority relation
σ	firing sequence
$\mathit{\Sigma}$	Petri net
φ	left part of a sequent
ψ	right part of a sequent

Main operators and functions

$enabled(M)$	set of transitions enabled in M
$M(p)$	number of tokens in place p or activity of local state p at M
$M(P)$	sum of tokens in places belonging to P at M
$[M\rangle$	set of markings or global states, reachable from M
$P(p)$	set of local states parallel to p
$V(G)$	set of nodes of graph G
$^\bullet x$	set of predecessors of node (place or transition) x of a Petri net
$x^\bullet$	set of successors of node (place or transition) x of a Petri net
$[x]$	cluster of Petri net containing node x
$\|\sigma\|$	length of firing sequence σ

Main abbreviations

BDD	binary decision diagram
CAD	computer aided design
CNF	conjunctive normal form
DNF	disjunctive normal form
EFC	extended free choice
FSM	finite state machine
HPN	hierarchical Petri net
IPN	interpreted Petri net
JPVM	Java Parallel Virtual Machine
LS	live and safe
OPN	operational Petri net
OPT	”optimal simulation”
PN	Petri net

PNSF	Petri Net Specification Format
PSS	"parallel selective search"
PDG	program dependency graph
RRG	reduced reachability graph
SCC	strongly connected component
SFC	Sequential Function Chart
SM	state machine
TC	terminal component
UML	Unified Modelling Language
XML	Extensible Markup Language

Lecture Notes in Control and Information Sciences

Edited by M. Thoma, M. Morari

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

Vol. 356: Karatkevich A.:
Dynamic Analysis of Petri Net-Based Discrete Systems
166 p. 2007 [978-3-540-71464-4]

Vol. 355: Zhang H.; Xie L.:
Control and Estimation of Systems with Input/Output Delays
213 p. 2007 [978-3-540-71118-6]

Vol. 354: Witczak M.:
Modelling and Estimation Strategies for Fault Diagnosis of Non-Linear Systems
215 p. 2007 [978-3-540-71114-8]

Vol. 353: Bonivento C.; Isidori A.; Marconi L.; Rossi C. (Eds.)
Advances in Control Theory and Applications
305 p. 2007 [978-3-540-70700-4]

Vol. 352: Chiasson, J.; Loiseau, J.J. (Eds.)
Applications of Time Delay Systems
358 p. 2007 [978-3-540-49555-0]

Vol. 351: Lin, C.; Wang, Q.-G.; Lee, T.H., He, Y.
LMI Approach to Analysis and Control of Takagi-Sugeno Fuzzy Systems with Time Delay
204 p. 2007 [978-3-540-49552-9]

Vol. 350: Bandyopadhyay, B.; Manjunath, T.C.; Umapathy, M.
Modeling, Control and Implementation of Smart Structures 250 p. 2007 [978-3-540-48393-9]

Vol. 349: Rogers, E.T.A.; Galkowski, K.; Owens, D.H.
Control Systems Theory and Applications for Linear Repetitive Processes 482 p. 2007 [978-3-540-42663-9]

Vol. 347: Assawinchaichote, W.; Nguang, K.S.; Shi P.
Fuzzy Control and Filter Design for Uncertain Fuzzy Systems
188 p. 2006 [978-3-540-37011-6]

Vol. 346: Tarbouriech, S.; Garcia, G.; Glattfelder, A.H. (Eds.)
Advanced Strategies in Control Systems with Input and Output Constraints
480 p. 2006 [978-3-540-37009-3]

Vol. 345: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Computing in Signal Processing and Pattern Recognition
1179 p. 2006 [978-3-540-37257-8]

Vol. 344: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Control and Automation
1121 p. 2006 [978-3-540-37255-4]

Vol. 341: Commault, C.; Marchand, N. (Eds.)
Positive Systems
448 p. 2006 [978-3-540-34771-2]

Vol. 340: Diehl, M.; Mombaur, K. (Eds.)
Fast Motions in Biomechanics and Robotics
500 p. 2006 [978-3-540-36118-3]

Vol. 339: Alamir, M.
Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes
325 p. 2006 [978-1-84628-470-0]

Vol. 338: Tokarzewski, J.
Finite Zeros in Discrete Time Control Systems
325 p. 2006 [978-3-540-33464-4]

Vol. 337: Blom, H.; Lygeros, J. (Eds.)
Stochastic Hybrid Systems
395 p. 2006 [978-3-540-33466-8]

Vol. 336: Pettersen, K.Y.; Gravdahl, J.T.; Nijmeijer, H. (Eds.)
Group Coordination and Cooperative Control
310 p. 2006 [978-3-540-33468-2]

Vol. 335: Kozłowski, K. (Ed.)
Robot Motion and Control
424 p. 2006 [978-1-84628-404-5]

Vol. 334: Edwards, C.; Fossas Colet, E.; Fridman, L. (Eds.)
Advances in Variable Structure and Sliding Mode Control
504 p. 2006 [978-3-540-32800-1]

Vol. 333: Banavar, R.N.; Sankaranarayanan, V.
Switched Finite Time Control of a Class of Underactuated Systems
99 p. 2006 [978-3-540-32799-8]

Vol. 332: Xu, S.; Lam, J.
Robust Control and Filtering of Singular Systems
234 p. 2006 [978-3-540-32797-4]

Vol. 331: Antsaklis, P.J.; Tabuada, P. (Eds.)
Networked Embedded Sensing and Control
367 p. 2006 [978-3-540-32794-3]

Vol. 330: Koumoutsakos, P.; Mezić, I. (Eds.)
Control of Fluid Flow
200 p. 2006 [978-3-540-25140-8]

Vol. 329: Francis, B.A.; Smith, M.C.; Willems, J.C. (Eds.)
Control of Uncertain Systems: Modelling, Approximation, and Design
429 p. 2006 [978-3-540-31754-8]

Vol. 328: Loría, A.; Lamnabhi-Lagarrigue, F.; Panteley, E. (Eds.)
Advanced Topics in Control Systems Theory
305 p. 2006 [978-1-84628-313-0]

Vol. 327: Fournier, J.-D.; Grimm, J.; Leblond, J.; Partington, J.R. (Eds.)
Harmonic Analysis and Rational Approximation
301 p. 2006 [978-3-540-30922-2]

Vol. 326: Wang, H.-S.; Yung, C.-F.; Chang, F.-R.
H_∞ Control for Nonlinear Descriptor Systems
164 p. 2006 [978-1-84628-289-8]

Vol. 325: Amato, F.
Robust Control of Linear Systems Subject to Uncertain
Time-Varying Parameters
180 p. 2006 [978-3-540-23950-5]

Vol. 324: Christofides, P.; El-Farra, N.
Control of Nonlinear and Hybrid Process Systems
446 p. 2005 [978-3-540-28456-7]

Vol. 323: Bandyopadhyay, B.; Janardhanan, S.
Discrete-time Sliding Mode Control
147 p. 2005 [978-3-540-28140-5]

Vol. 322: Meurer, T.; Graichen, K.; Gilles, E.D. (Eds.)
Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems
422 p. 2005 [978-3-540-27938-9]

Vol. 321: Dayawansa, W.P.; Lindquist, A.; Zhou, Y. (Eds.)
New Directions and Applications in Control Theory
400 p. 2005 [978-3-540-23953-6]

Vol. 320: Steffen, T.
Control Reconfiguration of Dynamical Systems
290 p. 2005 [978-3-540-25730-1]

Vol. 319: Hofbaur, M.W.
Hybrid Estimation of Complex Systems
148 p. 2005 [978-3-540-25727-1]

Vol. 318: Gershon, E.; Shaked, U.; Yaesh, I.
H_∞ Control and Estimation of State-multiplicative Linear Systems
256 p. 2005 [978-1-85233-997-5]

Vol. 317: Ma, C.; Wonham, M.
Nonblocking Supervisory Control of State Tree Structures
208 p. 2005 [978-3-540-25069-2]

Vol. 316: Patel, R.V.; Shadpey, F.
Control of Redundant Robot Manipulators
224 p. 2005 [978-3-540-25071-5]

Vol. 315: Herbordt, W.
Sound Capture for Human/Machine Interfaces: Practical Aspects of Microphone Array Signal Processing
286 p. 2005 [978-3-540-23954-3]

Vol. 314: Gil', M.I.
Explicit Stability Conditions for Continuous Systems
193 p. 2005 [978-3-540-23984-0]

Vol. 313: Li, Z.; Soh, Y.; Wen, C.
Switched and Impulsive Systems
277 p. 2005 [978-3-540-23952-9]

Vol. 312: Henrion, D.; Garulli, A. (Eds.)
Positive Polynomials in Control
313 p. 2005 [978-3-540-23948-2]

Vol. 311: Lamnabhi-Lagarrigue, F.; Loría, A.; Panteley, E. (Eds.)
Advanced Topics in Control Systems Theory
294 p. 2005 [978-1-85233-923-4]

Vol. 310: Janczak, A.
Identification of Nonlinear Systems Using Neural Networks and Polynomial Models
197 p. 2005 [978-3-540-23185-1]

Vol. 309: Kumar, V.; Leonard, N.; Morse, A.S. (Eds.)
Cooperative Control
301 p. 2005 [978-3-540-22861-5]

Vol. 308: Tarbouriech, S.; Abdallah, C.T.; Chiasson, J. (Eds.)
Advances in Communication Control Networks
358 p. 2005 [978-3-540-22819-6]

Vol. 307: Kwon, S.J.; Chung, W.K.
Perturbation Compensator based Robust Tracking Control and State Estimation of Mechanical Systems
158 p. 2004 [978-3-540-22077-0]

Vol. 306: Bien, Z.Z.; Stefanov, D. (Eds.)
Advances in Rehabilitation
472 p. 2004 [978-3-540-21986-6]

Vol. 305: Nebylov, A.
Ensuring Control Accuracy
256 p. 2004 [978-3-540-21876-0]

Vol. 304: Margaris, N.I.
Theory of the Non-linear Analog Phase Locked Loop
303 p. 2004 [978-3-540-21339-0]

Vol. 303: Mahmoud, M.S.
Resilient Control of Uncertain Dynamical Systems
278 p. 2004 [978-3-540-21351-2]

Vol. 302: Filatov, N.M.; Unbehauen, H.
Adaptive Dual Control: Theory and Applications
237 p. 2004 [978-3-540-21373-4]

¥860.00元

Contents

1. Introduction

Design of modern discrete devices and systems often deals with parallel processes and structures. For that reason practically all modern hardware design languages and formalisms used for system specification (such as VHDL, Verilog) allow describing concurrency. Design of complex, VLSI-based electronic devices is possible only with the help of CAD systems, so the design and verification methods have to be (and mostly are) formalized. Formalization and automatization of system design requires developing of formal models for parallel discrete systems and low-level description languages based on these models.

Specifications of devices and systems described in VHDL, Verilog or other popular languages of logical control, as LD, IL or ST [159], are very difficult for formal verification, because it is practically impossible to create adequate and at the same time simple formal models for such specifications (if these languages are used without restrictions). The problem can be solved by using of restricted specifications based on models which are easy to analyze and have enough expressive power.

There are two main directions of developing such models, each having its good and bad aspects. Both of them are, in a sense, extensions of the finite state machines (FSM) - the basic model of sequential discrete devices, which is, of course, in its "pure" version not convenient for practical needs of specifying of complex systems.

One direction is the composition of FSMs in various ways. The simplest implementation of this approach is the FSM network - a system of communicating automata [23, 147]. Studies on the automata networks have started in 1960-s, but rapid development of the methods of behavior specification by means of such networks began in 1980-s. Adding hierarchy to FSM networks leads to obtaining the model known as HCFSM (Hierarchical Concurrent Finite State Machines) [71]. One of the most popular and well-adapted to HCFSM languages has been developed within a frame of the universal specification language UML (Unified Modelling Language [209]), describing hierarchical objects and dependencies between them. We talk about the Statechars, invented by D. Harel [56, 83]. There exist several other models and languages based on automata networks such as

SMV [177], Promela [89, 90], CFSM [18], Requirements State Machine Language and visualState [200]. Probably the best implementation of this approach is the Ptolemy project, developed in Berkeley university [52, 157].

Another direction is the Petri nets and Petri net-based models and languages. A "pure" Petri net can describe a structure of parallel algorithm in convenient way, but it cannot describe interaction with the outer world (controlled objects). In order to develop Petri net models useful for discrete system design, the nets have to be enhanced at least by input and output signals. Often also such elements, as internal signals, time dependencies, operations on integers and other non-binary data are used. Probably the first successful attempt to create a Petri net-based language for control specification was the model known as GRAFCET. Its first version was developed in France by the working group called "Logical Systems" from AFCET (Association Française de Cybernétique Economique et Technique) in the 1970s [51, 205]. In 1988 it was adopted by the International Electrotechnical Commission as an international standard under the name of "Sequential Function Chart" (SFC)[159]. Translators have been developed to implement GRAFCET on programmable controllers. In the late 1970s and the early 1980s intensive researches in similar direction have started in the USSR (Institute of Engineering Cybernetics, Minsk [236, 238, 239, 240]; Institute of Control Sciences, Moscow [232, 233]) and in Poland (Technical University of Zielona Góra [3, 224]); parallel automata models [4, 240] and PRALU language [245, 249] arose, the methods of implementation and verification of such descriptions were designed[1]. Later various kinds of colored, interpreted, object and hybrid Petri nets and similar models have been developed, studied and applied (see e.g. [12, 34, 67, 94, 95, 100, 137, 143, 195, 196]). Most of these models allow hierarchical description, which is necessary for modern system design. In this case a net place or (more rarely) a net transition may be considered at lower level as a net.

These two approaches are equivalent in their expressiveness, each of them has its ardent supporters, the Petri net models and FSM-based models can be transformed into each other [148], and the question, which of them is "better" for system design, is still discussed. Both of them are used in CAD systems (see, for example, [56, 253]; however, Petri net models, being popular among the researchers, are definitively less popular among industrial CAD designers), and there is practical need to develop analysis methods for the models of both kinds.

A control system can be implemented using one or several microcontrollers, FPGA devices, specialized or general-purpose computers and so on. At the level of control algorithm specification and its verification there is no difference, which way of implementation will be used at further steps of design. So, in this book analysis and verification tasks are considered independently of the implementation details.

[1] These researches were preceded by studying the *sequent descriptions* [4, 80, 234, 237, 250], probably inspired by the theory of logical inference introduced by Gentzen [204]. Now sequent descriptions (sequent automata) are used as one of the intermediate specifications during implementation of parallel automata [5, 243, 248, 249].

1.1 Analysis of Parallel Discrete Systems

Methods of formal verification are a necessary part of any methodology of computer-aided design of hardware or software systems - and, in most cases, the verification is a bottleneck of the designs. Verification of parallel safety-critical systems differs from the verification of sequential designs, because there are some additional important properties, guaranteed or easy to check in case of sequential systems. The main conditions of a "good" parallel system are the next: [200, 245, 249]:

- lack of redundancy (lack of unreachable local states[2] and operations which are never executed);
- deadlock-freeness (in some cases the specified deadlocks must be reachable in a system; generally, detection of global and local deadlocks - situations, in which some or all parts of the system cannot react on input events because of mutual blocking - is one of the main analysis tasks) and a wider property, liveness (which implies lack of redundancy); often (for the cyclic systems) the condition of reversibility is added;
- safeness - no operation can be re-initialized during its execution;
- determinism - for parallel systems it additionally means, that parallel branches never destroy conditions and results of each other.

Of course, a designer may choose and formalize some specific conditions for specific designs. Checking of most conditions can be reduced to solving of reachability problem (reachability of a specified state or one of the states belonging to a specified class). Reachability is usually not a property difficult to check for a sequential system such as FSM, but for parallel systems the situation is different.

Analysis and formal verification of the parallel systems is a much more complex task than verification of a single FSM. The main problem is caused by the fact, that the parallel systems may have huge number of reachable states (it may depend exponentially on the of system size; for example, number of states of an FSM, equivalent to a parallel automaton with n states, may be maximally $3^{n/3}/n$ [249]. A parallel system may even have the infinite state space, such as an unbounded Petri net). That's why analysis by reducing of a parallel system to sequential one or by generating its state space in explicit form is practically impossible even for relatively simple systems (a parallel automaton with several dozens of local states may have milliards of global spaces). Model checking, a popular technology of formal verification based on state space analysis (using some tricks to prune the state space and, in most cases, representing state space in compact form like BDD), is practically used to verify properties of state spaces of size as large as about 1000 states maximum (accirding to [177]; however, in the earlier publication [45] the specific examples with an extremely large number of states - about 10^{16}, 10^{20} and even 10^{120} - are mentioned, successfully

[2] For parallel systems the global states (state of the whole system) and local states should be distinguished; formal definitions will be given later.