DIMACS

Series in Discrete Mathematics
and Theoretical Computer Science

Volume 32

The SPIN Verification System

The Second Workshop on the
SPIN Verification System
Proceedings of a DIMACS Workshop
August 5, 1996

Jean-Charles Grégoire
Gerard J. Holzmann
Doron A. Peled
Editors

American Mathematical Society

Selected Titles in This Series

(Continued from the front of this publication)

15 **Nathaniel Dean and Gregory E. Shannon, Editors**
Computational Support for Discrete Mathematics

14 **Robert Calderbank, G. David Forney, Jr., and Nader Moayeri, Editors**
Coding and Quantization: DIMACS/IEEE Workshop

13 **Jin-Yi Cai, Editor**
Advances in Computational Complexity Theory

12 **David S. Johnson and Catherine C. McGeoch, Editors**
Network Flows and Matching: First DIMACS Implementation Challenge

11 **Larry Finkelstein and William M. Kantor, Editors**
Groups and Computation

10 **Joel Friedman, Editor**
Expanding Graphs

9 **William T. Trotter, Editor**
Planar Graphs

8 **Simon Gindikin, Editor**
Mathematical Methods of Analysis of Biopolymer Sequences

7 **Lyle A. McGeoch and Daniel D. Sleator, Editors**
On-Line Algorithms

6 **Jacob E. Goodman, Richard Pollack, and William Steiger, Editors**
Discrete and Computational Geometry: Papers from the DIMACS Special Year

5 **Frank Hwang, Fred Roberts, and Clyde Monma, Editors**
Reliability of Computer and Communication Networks

4 **Peter Gritzmann and Bernd Sturmfels, Editors**
Applied Geometry and Discrete Mathematics, The Victor Klee Festschrift

3 **E. M. Clarke and R. P. Kurshan, Editors**
Computer-Aided Verification '90

2 **Joan Feigenbaum and Michael Merritt, Editors**
Distributed Computing and Cryptography

1 **William Cook and Paul D. Seymour, Editors**
Polyhedral Combinatorics

This page intentionally left blank

https://doi.org/10.1090/dimacs/032

DIMACS

Series in Discrete Mathematics
and Theoretical Computer Science

Volume 32

The SPIN Verification System

The Second Workshop on the
SPIN Verification System
Proceedings of a DIMACS Workshop
August 5, 1996

Jean-Charles Grégoire
Gerard J. Holzmann
Doron A. Peled
Editors

NSF Science and Technology Center
in Discrete Mathematics and Theoretical Computer Science
A consortium of Rutgers University, Princeton University,
AT&T Labs, Bell Labs, and Bellcore

American Mathematical Society

This DIMACS volume contains the proceedings of the second workshop on the SPIN Verification System held at DIMACS on August 5, 1996.

1991 *Mathematics Subject Classification.* Primary 68Q60; Secondary 68Q45, 68Q10, 68M10, 68N99, 68Q68, 68-06.

Library of Congress Cataloging-in-Publication Data

The SPIN Verification System (1996 : New Brunswick, N. J.)

The SPIN Verification System : DIMACS workshop, August 5, 1996 / Jean-Charles Grégoire, Gerard J. Holzmann, Doron A. Peled, editors.

p. cm. — (DIMACS series in discrete mathematics and theoretical computer science, ISSN 1052-1798 ; v. 32)

Workshop held at Rutgers Univ. in New Brunswick, N. J.

Includes bibliographical references.

ISBN 0-8218-0680-7

1. Computer software—Verification—Congresses. 2. SPIN (Computer file)—Congresses. I. Grégoire, Jean-Charles, 1960– . II. Holzmann, Gerard J., 1951– . III. Peled, Doron, 1962– . IV. Title. V. Series.

QA76.76.V47W677 1996

005.2′76—dc21 96-54839

CIP

---

10 9 8 7 6 5 4 3 2 1 02 01 00 99 98 97

CONTENTS

Foreword

The Second SPIN Workshop, held at Rutgers University on August 5, 1996, was part of DIMACS Special Year on Logic and Algorithms. We would like to express our appreciation to Jean-Charles Grégoire, Gerard J. Holzmann, and Doron A. Peled for their efforts to organize and plan the workshop.

The workshop was part of the broader Special Year on Logic and Algorithms program which focused on computer aided verification, finite models, and proof complexity. The special year encouraged collaborations among very different research communities and this volume records one of many workshops in which this was achieved. We extend our thanks to Eric Allender, Robert Kurshan, and Moshe Vardi for their work over many months as special year organizers.

DIMACS gratefully acknowledges the generous support that makes these programs possible. The National Science Foundation, through its Science and Technology Center program, the New Jersey Commission on Science and Technology, DIMACS partners at Rutgers, Princeton, AT&T Labs, Bell Labs, and Bellcore generously supported the special year. Additional funding from Bell Labs allowed increasing the number of scientists who could participate.

Fred S. Roberts
Director

Bernard Chazelle
Co-Director for Princeton

Stephen R. Mahaney
Associate Director

This page intentionally left blank

PREFACE

What Is SPIN? SPIN is a general tool for the specification and formal verification of software for distributed systems. It has been used to detect design errors in a wide range of applications, such as abstract distributed algorithms, data communications protocols, operating systems code, and telephone switching code. The verifier can check for basic correctness properties, such as absence of deadlock and race conditions, logical completeness, or unwarranted assumptions about the relative speeds of processes. It can also check for more subtle, system dependent, correctness properties expressed in the syntax of Linear-time Temporal Logic [**14**]. The tool translates LTL formulae automatically into automata representations [**3**], which can be used in an efficient on-the-fly verification procedure.

Some Background. Work on the construction of automated verification systems of this type started about two decades ago. Among the first to built a fully automated tool based on the reachability analysis of finite state models was Jan Hajek at the Technical University in Eindhoven in The Netherlands [**4**]. Between 1976 and 1978 Hajek's system *Approver* successfully uncovered bugs in a series of published designs for communications protocols. The algorithmic techniques on which the *Approver* system was based, were unfortunately never revealed, and therefore the system could only inspire, but not directly influence related efforts by others.

At approximately the same time, Colin West at the IBM research lab in Rüschlikon, Switzerland, worked on the implementation of a tool for Pitro Zafiropulo's duologue matrix analysis technique. This work quickly lead West to develop his own variant of a verification system [**18**]. The most visible result of this work was a first verification, and the uncovering of design flaws, in the X.21 recommendation from the CCITT (now the ITU) with West's perturbation analysis procedure. The X.21 verification is today frequently used as a litmus test for new verification systems.

The work that ultimately lead to the SPIN verification system started at Bell Labs in 1980. The first incarnation of the system, the verifier PAN, started finding bugs in data-switch control protocols in November 1980. Like today's SPIN, PAN was a general on-the-fly verification system, but unlike SPIN it was restricted to the verification of only basic safety properties. Over a period of ten years [**5, 6, 7, 8, 9**], this tool evolved into a powerful verification system with full model checking capabilities. SPIN was first released for general distribution in late 1990 [**10**], and has continued to evolve. Significant improvements in SPIN's model checking capabilities were the introduction of a partial order reduction method in 1994 [**13, 11**] and a built-in translation algorithm [**3**] for converting LTL formulae into the automata recognized by SPIN's verification engine. The code to the SPIN system is available from `http://netlib.bell-labs.com/netlib/spin/whatispin.html`.

Several other tools with a similar long history exist. Descriptions can be found, e.g., in [**1, 15, 16, 12**].

Tool Characteristics. SPIN has several distinguishing features that make it well suited for addressing verification problems in the general area of concurrent software design, and telecommunications systems engineering:

- The specification language for SPIN is a high-level, *asynchronous* and non-deterministic, guarded command language, that is well suited for specifying software process behaviors, instead of a synchronous notation that would be better suited for specifying hardware circuits.
- The logic used in SPIN is based on the linear-time temporal logic LTL, instead of a branching time temporal logic, such as CTL.
- The verification procedure used in SPIN is based on explicit state enumeration, rather than on a symbolic state representation.
- SPIN uses an on-the-fly verification algorithm [**2**], storing as little information in memory as is strictly necessary for completing the verification task, instead of an offline (or two-pass) verification procedure. No transitions (edges) need be stored with this method, and efficient compression and reduction techniques are available to reduce the memory usage to a minimum without incurring undue overhead, or unpredictable performance.
- SPIN uses a general partial order reduction technique [**13, 11**] to exploit regularities that are common in asynchronous interleaving systems, rather than the BDD representations that are common in hardware verification to achieve the same effects.
- SPIN contains a range of simulation options, with either graphical or textual output, that have proven their value in the pre-verification phase of a design. A simple graphical user interface to the system, called XSPIN, enhances the usability of the tool especially to new or occasional users.

The Spin Workshops. This book contains the proceedings of the second workshop on work related to the SPIN verification system. At the workshop, fourteen research papers were presented by researchers from eight different countries.

The keynote presentation was delivered by Moshe Vardi, Noah Harding Professor and chair of Computer Science at Rice University. Prof. Vardi is one of the primary developers of the automata theoretic framework on which SPIN is founded [**17**]. The viewgraphs of the keynote presentation are available as an online document at `http://netlib.bell-labs.com/netlib/spin/ws96/vardi.ps`. One other presentation could not be included in these proceedings as a full paper: work reported by Frank Schneider and Jack Callahan from NASA's Software Verification and Validation Facility, on the verification with SPIN of aspects of a distributed system used in one of NASA's upcoming space missions. We have added a paper to this volume that was also distributed as part of the participants proceedings to the SPIN workshop: an outline for an operational semantics definition of SPIN's specification language PROMELA.

Four tool demos were presented at the workshop:

1. The WHEEL environment for SPIN: an extension targeted to the specification and verification of feature interaction problems (by F.J. Lin, Bellcore, USA).
2. A real-time extension of SPIN (by Stavros Tripakis, Verimag, France).
3. A PROMELA to C translator (by Siegfried Loeffler, Hewlett-Packard, England).

4. RSPIN, an extension for the specification and verification of reactive systems (by Frank Olsen, Centre National d'Etudes des Telecommunications, Issy-Les-Moulineaux, France).

The number of places where the SPIN system is installed and used now numbers in the thousands. The program of this year's workshop reflects the nature of the work in formal verification that is triggered or inspired by this system. In four broad categories, there are:

- Theoretical and foundational studies.
- Empirical studies of the relative effectiveness of different types of search and storage algorithms.
- Significant practical applications.
- Extensions and revisions of the basic SPIN code.

Several of the projects in each category were represented at the workshop, but perhaps more encouraging still is that many more of these SPIN projects are in progress at academic and industrial research labs around the world. The goal of the SPIN workshops is to create an opportunity for those who work with this system to meet, share experiences, learn about each others work, and exchange ideas. As witnessed by these proceedings, this year's workshop fully met that goal.

Acknowledgements. The 1996 workshop was sponsored by Bell Labs, and by DIMACS, the National Science Foundation's Science and Technology Center for Discrete Mathematics and Theoretical Computer Science, as part of their Special Year Program on *Logic and Algorithms.* DIMACS provided both logistic support at the workshop location at Rutgers University in New Brunswick, New Jersey, and provided financial support for invited speakers and graduate students. We are especially grateful to Pat Pravato, Sarah Donnelly, Wanglai Li, and Hangbiao Shi for their courteous and efficient help with the preparations, often under time pressure of preparations for other special year events.

Jean-Charles Grégoire, Gerard J. Holzmann, Doron Peled.

References

[1] E. M. Clarke and E. A. Emerson, Characterizing properties of parallel programs as fixpoints, *Proc. 7th Int. Coll. on Automata, Languages and Programming*, LNCS 85, 1981.

[2] C. Courcoubetis, M. Vardi, P. Wolper, M, Yannakakis, Memory-efficient algorithms for the verification of temporal properties, Formal methods in system design 1 (1992) 275–288.

[3] R. Gerth, D. Peled, M. Vardi, P. Wolper, Simple on-the-fly automatic verification of linear temporal logic, *Proc. 15th Int. Conf. on Protocol Specification, Testing, and Verification*, INWG/IFIP, Eds. P. Dembinsky, and M. Sredniawa, Warsaw Poland, 1995.

[4] J. Hajek, Automatically verified data transfer protocols, *Proc. 4th ICCC*, 1978, Kyoto, pp. 749-756.

[5] G. J. Holzmann, *PAN - a Protocol Specification Analyzer*, Bell Laboratories Technical Memorandum, TM81-11271-5, May 1981.

[6] G. J. Holzmann, R. A. Beukers, The PANDORA protocol development system, *Proc. 3rd Int. Conf on Protocol Specification, Testing, and Verification*, INWG/IFIP, Eds. H. Rudin and C. West, pp. 357–369, North Holland Publ. Co., June, 1983.

[7] G. J. Holzmann, Tracing protocols, *AT&T Techn. Journal*, Vol 64, No. 10, Dec. 1985.

[8] G. J. Holzmann, An improved reachability analysis technique, *Software Practice and Experience*, Vol. 18, No. 2, pp. 137-161, Feb. 1988.

[9] G. J. Holzmann, J. Patti, Validating SDL specifications: An Experiment, *Proc. 9th Int. Conf on Protocol Specification, Testing, and Verification*, INWG/IFIP, Ed. C. Vissers and E. Brinksma, Twente, Neth., June, 1989.

[10] G. J. Holzmann, *Design and Validation of Computer Protocols*, Prentice Hall, 1991.

[11] G. J. Holzmann and D. Peled, An improvement in formal verification, *Proc. 7th Int. Conf. on Formal Description Techniques*, Eds. D. Hogrefe and S. Leue, FORTE94, Berne, Switzerland. October 1994.

[12] R. P. Kurshan, Computer Aided Verification of Coordinating Processes, Princeton University Press, 1994.

[13] D. A. Peled, Combining partial order reductions with on-the-fly model checking, *Proc. 6th Int. Conf. on Computer Aided Verification*, CAV94, Stanford, Ca., June 1994.

[14] A. Pnueli, The temporal logic of programs, *Proc. 18th IEEE Symposium on Foundations of Computer Science*, Providence, R.I., pp. 46-57, 1977.

[15] J. P. Queille, *Le systéme César: description, spécification et analyse des applications réparties*, Ph.D. Thesis, June 1982, Computer Science Dept., Univ. Grenoble, France.

[16] K.L. McMillan, *Symbolic model checking: an approach to the state explosion problem*, Kluwer Academic Publ., 1993.

[17] M.Y. Vardi and P. Wolper, An automata-theoretic approach to automatic program verification, *Proc. First Symposium on Logic in Computer Science*, June 1986, Cambridge, pp. 322–331.

[18] C. H. West, General technique for communications protocol validation, *IBM Journal of Research and Development*, Vol 22, No. 4, p. 393, 1978.

ISBN 0-8218-0680-7

9 780821 806807