Advanced SPIN

Proposal for a half-day tutorial at SPIN 2004

Theo C. Ruys[1] and Gerard J. Holzmann[2]

[1] Department of Computer Science, University of Twente.
P.O. Box 217, 7500 AE Enschede, The Netherlands.
`http://www.cs.utwente.nl/~ruys/`
[2] NASA/JPL, Laboratory for Reliable Software.
4800 Oak Grove Drive, Pasadena, CA 91109, USA.
`http://spinroot.com/gerard/`

Abstract. SPIN [2, 11] is a model checker for the verification of distributed systems software. The tool is freely distributed, and often described as one of the most widely used verification systems. [2] describes SPIN 4.0, the latest version of the tool. SPIN was awarded the ACM Software System Award for 2001 [1]. Advanced SPIN is a 'sequel' tutorial to [9] and is targeted towards intermediate to advanced SPIN users.

The tutorial starts with a brief overview of the latest additions to PROMELA, the specification language of SPIN. General patterns are discussed to contruct efficient PROMELA models and how to use SPIN in the most effective way. Topics to be discussed include: SPIN's optimisation algorithms, directives and options to tune verification runs with SPIN and guidelines for effective PROMELA modelling (e.g. invariance, atomicity, time, lossy channels, scheduling, etc.).

The second part of the tutorial looks in more detail at the theoretical underpinnings of SPIN, and discusses some of its more recent applications to the verification of implementation level systems code, using model extraction techniques. Also basic and more advanced abstraction techniques for building SPIN models will be discussed, and some examples of large applications of SPIN based logic model checking. Topics to be discussed include: automata theoretic verification, model construction, abstraction and extraction, application studies, etc.

1 Introduction

SPIN [2–6, 11] supports the formal verification of distributed systems code. The software was developed at Bell Labs in the formal methods and verification group starting in 1980. SPIN is freely distributed, and often described as one of the most widely used verification systems (estimation: between 5,000 and 10,000 people routinely use SPIN). The SPIN software is written in ANSI standard C, and is portable across all versions of the UNIX operating system. It can also be compiled to run on any standard PC running Linux or Windows 95/98/NT/2000/ME/XP.

The automata-theoretic foundation for SPIN is laid by [12]. The very recent [2] describes SPIN 4.0, the latest version of the tool. SPIN was awarded the ACM Software System Award for 2001 [1].

2 Target audience

The “Advanced SPIN” tutorial targets intermediate and advanced SPIN users; users should have at least some experience with verification with SPIN.

General patterns are discussed to contruct efficient PROMELA models and how to use SPIN in the most effective way. The tutorial will also look under the hood of SPIN and discusses more of the theory and algorithms that make the tool work. Furthermore, the tutorial will discuss abstraction techniques on how to extract (PROMELA) models from implementation code. Problems related to the verification of industrial size systems will also be covered.

3 Aims and objectives

The objective of Advanced SPIN is to (further) educate the SPIN 2004 attendees to on model checking technology in general and SPIN in particular. After the tutorial, attendees should:

- be able to construct efficient and effective PROMELA models;
- be able to formulate effective properties that can be checked with SPIN;
- realise how model checking technology can be applied to software engineering;
- have a basic understanding of the theory and algorithms that make SPIN work efficiently;
- have a detailed understanding of the range of complexity management techniques that are supported by the tool;
- have a good understanding of the importance of abstraction in model construction;
- understand how and when verification models can be extracted from implementation level source code.

4 Duration, requirements and material

- Advanced SPIN will be a *half day* (3 to 4 hour) tutorial.
- The instructors rely on a *beamer* for the presentation of the material.
- The tutorial will *not* be accompanied by a survey paper (but [2] and [8] are available). Handouts of the transparencies of the tutorial will be provided, of course.

5 Summary of material

- Internals of SPIN. Before learning how to write efficient PROMELA models, one has to have a fealing of how SPIN works.

  state space, state vector, search depth, SPIN *optimization algorithms,* PROMELA *annotations, command-line* SPIN *options, compiler directives.*

- Effective PROMELA Patterns.
  One of the problems of formal verification is the so-called *state space explosion*. Several guidelines for effective PROMELA modelling and how to tune SPIN will be discussed.
  `bit`-vectors, terminating processes, invariance, modelling time, lossy channels, optimization problems [10], local and global variables.
- New SPIN 4.x Features.
  embedded C code, breadth-first-search
- Automata Theoretic Verification.
  labelled transition systems, accepting automata, Büchi automata, checking for emptiness, depth-first search.
- Model construction and Abstraction.
  building tractable models, design verification, reversing sources of computational complexity.
- Model extraction.
  static analysis, slicing, data flow analysis, abstraction rules, abstraction function.
- Application Studies.
  verification of call processing code of the PathStar switch, verification of flight control software for NASA's Cassini mission, verification of Deep Space 1 code.

6 Biographies

Theo Ruys is an assistant professor in the Formal Methods and Tools group at the University of Twente in the Netherlands, where he is responsible for courses in Programming and Compiler Construction. In 2001, he received his PhD under supervision of professor Ed Brinksma. His PhD Thesis "Towards Effective Model Checking" [8] discusses methods and techniques to improve the effectiveness of the model checking process with the goal to reliably apply model checking technology 'in the large'. A considerable part of this thesis is devoted to the model checker SPIN, where several techniques are discussed to use SPIN in the most effective way [7]. In April 2002, Theo Ruys presented a successful "SPIN Beginner's Tutorial" at the SPIN 2002 Workshop in Grenoble [9].

Gerard Holzmann is Principal Computer Scientist at NASA/JPL, Laboratory for Reliable Software, Pasadena, California. Formerly, he was Director of the Computing Principles Research Department at Bell Labs, in Murray Hill, New Jersey. He received his PhD in technical sciences in 1979 from Delft University in the Netherlands, and joined Bell Labs shortly thereafter. He does research in digital image processing, requirements engineering, software testing, distributed systems design, and computer aided verification. Gerard Holzmann has given numerous (invited) presentations on SPIN, including extensive tutorials, e.g., at the Marktoberdorf Summer school in 2000 and at the BRICS Autumn school in Verification in 1996.

References

1. ACM. ACM Software System Awards, URL: http://www.acm.org/awards/ssaward.html.
2. G. J. Holzman. *The SPIN Model Checker – Primer and Reference Manual*. Addison-Wesley, Boston, USA, 2004.
3. G. J. Holzmann. *Design and Validation of Computer Protocols*. Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.
4. G. J. Holzmann. Tutorial: Design and Validation of Protocols. *Computer Networks and ISDN Systems*, 25(9):981–1017, 1993.
5. G. J. Holzmann. SPIN Model Checking - Reliable Design of Concurrent Software. *Dr. Dobb's Journal*, pages 92–97, October 1997.
6. G. J. Holzmann. The Model Checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997.
7. T. C. Ruys. Low-Fat Recipes for SPIN. In K. Havelund, J. Penix, and W. Visser, editors, *SPIN Model Checking and Software Verification, Proc. of the 7th Int. SPIN Workshop (SPIN'2000)*, volume 1885 of *Lecture Notes in Computer Science (LNCS)*, pages 287–321, Stanford, California, USA, August 2000. Springer, Berlin.
8. T. C. Ruys. *Towards Effective Model Checking*. PhD thesis, University of Twente, Enschede, The Netherlands, March 2001. *Available from the author's homepage*.
9. T. C. Ruys. SPIN Tutorial: How to become a SPIN Doctor. In D. Bosnacki and S. Leue, editors, *Model Checking of Software, Proc. of the 9th Int. SPIN Workshop (SPIN'2002)*, volume 2318 of *Lecture Notes in Computer Science (LNCS)*, pages 6–13, Grenoble, France, April 2002. Springer, Berlin.
10. T. C. Ruys. Optimal Scheduling Using Branch and Bound with SPIN 4.0. In T. Ball and S. K. Rajamani, editors, *Model Checking of Software, Proc. of the 10th Int. SPIN Workshop (SPIN 2003)*, volume 2648 of *Lecture Notes in Computer Science (LNCS)*, pages 1–17, Portland, Oregon, USA, May 2003. Springer, Berlin.
11. SPIN Homepage. URL: http://spinroot.com/spin/.
12. M. Y. Vardi and P. Wolper. An Automatic-Theoretic Approach to Automatic Program Verification. In *Proc. of the First IEEE Symposium on Logic In Computer Science (LICS'86)*, pages 322–331, Cambridge, UK, June 1986.