Advanced SPIN Tutorial

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 [9] 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. The Advanced SPIN Tutorial is a *sequel* to [7] and is targeted towards intermediate to advanced SPIN users.

1 Introduction

SPIN [2–5, 9] 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. It is estimated that between 5,000 and 10,000 people routinely use SPIN. SPIN was awarded the ACM Software System Award for 2001 [1].

The automata-theoretic foundation for SPIN is laid by [10]. The very recent [5] describes SPIN 4.0, the latest version of the tool.

The SPIN software is written in standard ANSI C, and is portable across all versions of the UNIX operating system, including Mac OS X. It can also be compiled to run on any standard PC running Linux or Microsoft Windows.

2 Tutorial

The Advanced SPIN Tutorial is a *sequel* to [7] and is targeted towards intermediate to advanced SPIN users. The objective of the Advanced SPIN Tutorial is to (further) educate the SPIN 2004 attendees on model checking technology in general and SPIN in particular.

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 [6]. 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, modelling time, lossy channels, fairness, optimisation problems [8].

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 presented, 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, and application studies.

After the tutorial, attendees should:

- be able to construct (more) efficient and effective PROMELA models;
- be able to formulate effective properties that can be checked with SPIN;
- have a basic understanding of the theory and algorithms that make SPIN work efficiently;
- 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.

References

1. ACM Software Systems Awards. URL: http://www.acm.org/awards/ssaward.html.
2. G. J. Holzmann. *Design and Validation of Computer Protocols.* Prentice Hall, Englewood Cliffs, New Jersey, USA, 1991.
3. G. J. Holzmann. SPIN Model Checking - Reliable Design of Concurrent Software. *Dr. Dobb's Journal*, pages 92–97, October 1997.
4. G. J. Holzmann. The Model Checker SPIN. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997.
5. G. J. Holzmann. *The SPIN Model Checker – Primer and Reference Manual.* Addison-Wesley, Boston, Massachusetts, USA, 2004.
6. T. C. Ruys. *Towards Effective Model Checking.* PhD thesis, University of Twente, Enschede, The Netherlands, March 2001. *Available from the author's homepage.*
7. 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 *LNCS*, pages 6–13, Grenoble, France, April 2002.
8. 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 *LNCS*, pages 1–17, Portland, Oregon, USA, May 2003.
9. SPIN Homepage. URL: http://spinroot.com/spin/.
10. 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.