Software Engineering and Applications 软件工程与应用, 2014, 3, 70-77
Published Online June 2014 in Hans. http://www.hanspub.org/journal/sea
http://dx.doi.org/10.12677/sea.2014.33009

Hans汉斯

Research and Application of the SPIN Model Checker Graphical Tools

Ping Chen, Wangde Cheng

Anhui Xinhua University, Hefei
Email: chenping19891120@163.com

Received: Apr. 6th, 2014; revised: May 8th, 2014; accepted: May 17th, 2014

Abstract

Model checking is an automated formal analysis method which can provide counter-examples to verify system properties. It consists of system model, system described attributes and model checker. Model checker accepts model and attributes to verify whether the property meets the model automatically. If it is not satisfied, a counter-example will be given. Hardware and software verification have been widely used. SPIN model checker has been developed by Bell Labs, which is suitable for concurrent systems model checking tools. This paper describes the design and structure of SPIN, analyzes the theoretical basis and elaborates the installation & use of SPIN model checker and its graphical interface, iSpin, by specific examples. SPIN Promela language accepts the model established by the Promela language and the nature described by using linear temporal logic (LTL); hence, the results are verified. Based on the study of SPIN's structure and principle, a detailed exposition of the installation and use of its graphical interface iSpin is given.

Keywords

Model Checking, SPIN, Promela, LTL

模型检测器SPIN图形化工具的研究与应用

陈　平，王德成

安徽新华学院，合肥
Email: chenping19891120@163.com

收稿日期：2014年4月6日；修回日期：2014年5月8日；录用日期：2014年5月17日

摘　要

模型检测是一种有着自动化并能提供反例的验证系统属性的形式化分析方法，其由系统模型，系统属性的描述和模型检测器组成。模型检测器接受模型和属性自动的去验证属性是否满足模型，如果不满足给出一个反例，其在软硬件验证中都得到了广泛的应用。模型检测器SPIN是由贝尔实验室开发的一种适合于并发系统的模型检测工具。本文在介绍了SPIN的设计和结构，分析了其理论基础，并通过具体例子阐述了安装和使用模型检测器SPIN和它的图形化界面iSpin的使用方法。SPIN接受Promela语言建立的模型和使用线性时序逻辑LTL描述的性质，给出验证结果。本文在研究模型检测工具SPIN的结构和原理的基础上，详细的阐述了其图形化界面iSpin的安装和使用。

关键词

模型检测，SPIN，Promela，LTL

1. 引言

随着软硬件的快速发展，个人电脑、数码产品、各类网站等已经成为人们生活不可缺少的一部分，而这些设备中的软硬件系统日益复杂化，如何保证软硬件及通信系统的正确性和可靠性已经非常重要。在众多的方法中，模型检测是一种有效的形式化验证方法，其在设计阶段检测软硬件是否工作正确[1]。

形式化方法是用数学和逻辑的方法来验证软硬件系统，其由系统或者程序的描述和性质的描述组成。模型检测主要的工作是建模和形式化推理，其主要思想是采用二元的思想和手段对系统的一致性、安全性等属性或系统的性质进行推理和证明。其中建模是对系统的行为规约进行描述，再用某种逻辑表示系统的静态性质，最后检查系统行为与要求是否一致。其主要方法是对状态空间进行穷尽搜索，在搜索终止时，对于不满足的性质能给出反例[2]。随着模型检测技术的发展，产生了很多的模型检测工具，如 SPIN、SMV、NuSMV 等，其中 SPIN 是其中应用比较广泛的，适用于并行系统，并且在分布式系统、软件的形式化验证、协议的验证等方面取得了成功[3]。

模型检测器 SPIN 支持设计和验证系统，它通过建模语言 promela 来直观、明白地描述设计，使用线性时序逻辑 LTL 来有力地、简明地描述系统需求，并通过有效的算法验证系统是否满足需求[4]。SPIN 支持随机、交互和指导性的自动机验证，通过模仿系统的执行产生 C 程序，探测系统的状态空间。SPIN 验证主要关心的是进程之间的信息交互是否正确，而不是进程内部的具体计算，所以它的建模方式是：首先定义进程模板，每个进程模板作为一类进程的行为规范，实际系统可以看作是一个或若干进程模板实例的异步组合。

SPIN 是一个基于计算机科学的形式化方法，其将先进的理论验证方法应用在大型负责软件系统的验证当中，现在 SPIN 被广泛的应用在工业和学术界。本文在研究模型检测工具 SPIN 的结构和原理的基础上，详细的阐述了其图形化界面 iSpin 的安装和使用。

2. SPIN 原理和结构

2.1. SPIN 的原理

SPIN 是针对多线程软件的有效的检测工具，其主要检测进程之间的信息交互是否正确。SPIN 从一个并发系统的高规格的描述开始，在分析没有语法错误之后，对系统进行交互模拟，直到确认系统拥有预期的设计行为，然后，SPIN 产生一个 on-the-fly 验证程序，经检测器编译后被执行，在执行中如果发

现了违背正确性说明的任何反例，将这些反馈给交互模拟机并通过重现细节来剔除引起错误的原因[5]。

Promela 建模语言描述的系统由一个或更多的进程模板组成，每个进程模板描述各个进程的行为，作为一类进程的行为规范，并能通过进程模板创建更多的进程。如通过关键词 proctype 开始说明进程。SPIN 采用 on-the-fly 机制来构建自动机模型，将各个进程模板转换成一个有穷自动机，并发系统的全局行为通过计算自动机的异步交叉积得到的也少一个自动机。如果用 A 和 S 分别表示系统行为的集合所对应的自动机和待检测性质构造的自动机，L(A)和 L(S)分别表示 A 和 S 所接收的语言，其中 L(A)表示系统行为的集合，L(S)表示系统允许行为的集合。模型检测需要验证系统 A 是否满足规范 S，即 L(A) ∈ L(S)是否成立，被建模系统的行为集合是否都包含在性质所允许的行为集合中。那么系统性质所不允许的行为集合为语言 L(S)的补，记作 $\overline{L(S)}$，那么系统验证就是要检测性质规范 S 不允许的行为系统 A 是不会做的，即 $L(A)\cap\overline{L(S)}=\phi$，如果交集不为空，那么其中任何一个行为都可以作为反例。

一般情况下，求一个自动机的补自动机是比较困难的，这里利用 LTL 可以避免求自动机的补。首先将 LTL 公式转换成它的 NNF 形式，若 LTL 公式是 f，公式 f 的补为~f 即为不满足系统要求的性质，构造一个对应的 Büchi 自动机 $S_{\sim f}$ 来识别~f，这样检测 $L(A)\cap\overline{L(S)}=\phi$ 是否成立就可以转换为检测 $L(A)\cap L(S_{\sim f})=\phi$ 是否成立。具体的做法是检查是否存在一个从初始状态可达的环路包含一个至少一个接受状态来检查自动机是否为空，如果不满足，生成的自动机组合中必然存在一个可接受回路，这个回路以动作路径的方式给出反例。

在检测自动机的交是否为空时，SPIN 采用算法深度优先算法遍历自动机，并采用偏序简化技术来消除冗余。

2.2. SPIN 基本结构

SPIN 首先接受 Promela 语言描述的系统，经分析没有语法错误后，对系统进行交互模拟来验证系统的行为是否符合预期。然后，SPIN 从系统的高级规约中生成一个优化的 on-the-fly 验证程序，经检验器编译后被执行，执行中如果发现了任何反例，就回到交互模拟执行再继续探测，确定产生反例的原因。SPIN 的基本机构如图 1 所示，其描述了整个的检测过程[6]。

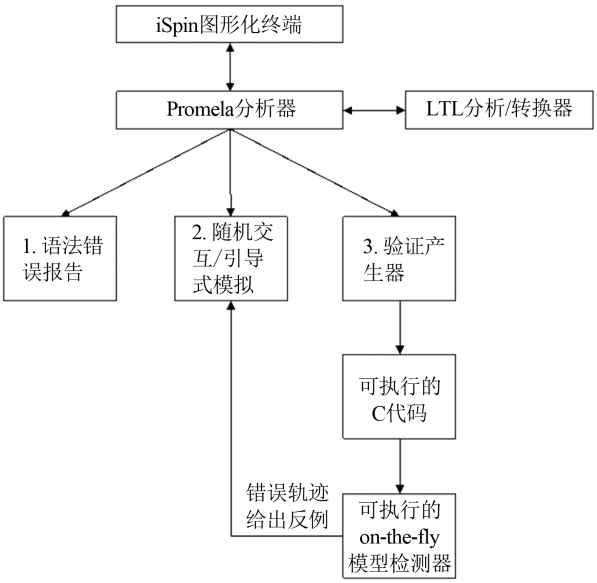

Figure 1. The basic structure of SPIN
图 1. SPIN 的基本结构图

3. 建模语言 Promela

Promela(Protocol/Process Meta Language)是一种用来对有限的状态系统建模的形式描述语言，其类似 C 程序语言，允许动态创建并行进程，在进程之间使用消息通道进行同步或异步的通信。因此，一个模型由进程、消息通道、全局对象、变量组成[7]。

Promela 语言强调模型进程间的通信，并且在一个时刻只能有一个状态发生变化。Promela 语言有丰富的进程交互原语，非确定性控制机构等主流语言没有的特点，能够构建高水平的分布式系统模型[8]。

4. 线性时序逻辑 LTL 及其到自动机的转换

线性时序逻辑 LTL 是一种形式化符号，能描述系统在执行路径上的性质，还可以表示系统的安全性、活性等。一个 LTL 公式由原子命题 p 和一元、二元、布尔、时序运算符的组合而成[9]。其语法如下：

```
f      ::=   p
             |true
             |false
             |(f)
             |f binop f
             | unop f
unop ::=   []      (总是)
             | <>     (有时)
             | ！     (逻辑非)
binop ::=  U       (直到)
             | &&   (逻辑与)
             | ||    (逻辑或)
             | ->        (实现)
             |  <->      (等价)
```

如上所说，SPIN 将 LTL 公式转换为自动机，例如，LTL 公式[](p U q)表示在 q 是真的之前，p 一直保持为真。该公式用 PROMELA 语言表示如图 2 所示。

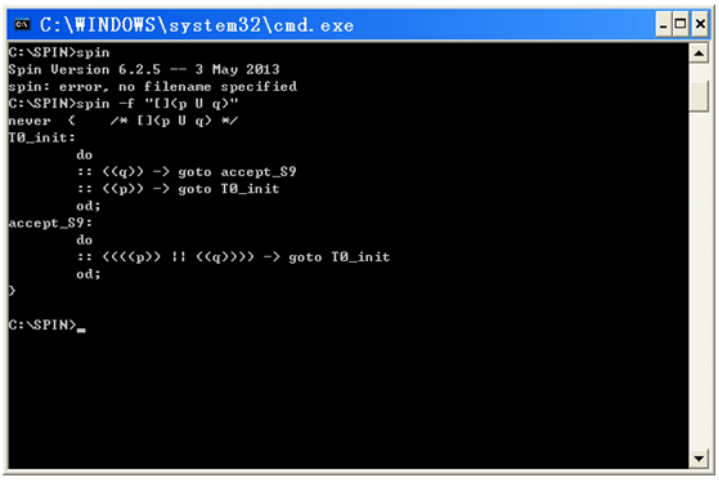

Figure 2. LTL formula syntax Promela representation
图 2. LTL 公式的 Promela 语法表示

可以看出自动机包含一个初始状态 T0_init 和一个接受状态 accept_S9，那么该 LTL 公式对应的状态机如图 3 所示。

5. 模型检测器 SPIN 的安装和 iSpin 的使用

SPIN 支持很多平台，如 Unix\Linux 系统、Windows、Macs 等，本文在 windows 环境安装 Spin 和 iSpin 有以下步骤[9]-[11]：

1) 安装 C 编译器 MinGW，如 D:\MinGW；
2) 下载 pc_spin*.zip 的最新 windows 版并解压到如 C:\ispin;
3) 下载安装 Tcl/Tk；
4) 安装 Graphviz；
5) 配置系统环境变量，如：“D:\MinGW\bin\;C:\spin;
6) C:\Tcl\bin;C:\ProgramFiles\Graphviz2.34\bin”；
7) 安装成功，双击 C:\ispin 中的 ispin.tcl 来运行 Spin 和 ispin。

6. SPIN 应用研究

SPIN 的应用非常广泛，可以检测协议、验证算法的正确性等。下面详细的描述了使用 iSpin 验证了 Dolev，Klawe 和 Rodeh 提出的无向环上的选举算法[12]。根据算法的描述，其模型描述为：

```
#define N 5 /* number of processes in the ring */
#define L 10 /* 2xN */
byte I;
mtype = { one, two, winner };
chan q[N] = [L] of { mtype, byte};
proctype nnode (chan inp, out; byte mynumber)
{bit Active = 1, know_winner = 0;
byte nr, maximum = mynumber, neighbourR;
xr inp;
xs out;
.....
}
init {
byte proc;
byte Ini[6];   /* N<=6 randomize the process numbers */
atomic {
I = 1;    /* pick a number to be assigned 1..N */
```

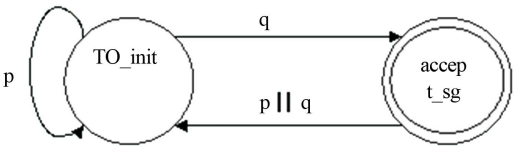

Figure 3. LTL formula [] (p U q) state machine
图 3. LTL 公式[] (p U q)的状态机

```
do
.....
od
}
}
```

首先，在 iSpin 中打开 leader.pml，其使用 promela 语言描述了 Dolev，Klawe 和 Rodeh 提出的无向环上的选举算法。同时，我们列出了使用 ltl 语言描述的一些即将被验证的属性。如：p0，p1，p2，p3，p4. 下面我们将通过具体的演示 iSpin 的使用。

iSpin 提供了很多标签，如 Edit/View，Simulate/Replay，Verification 等等。每个标签下面又提供了很多选择。如图 4 显示了标签 Edit/View 下面的功能有 Syntax Check，Redundancy Check 等。

通过在 Edit/view 标签中的 Syntax Check，Redundancy Check 之后，通过 Simulate/Replay 标签有(Re)Run，Stop 和 Rewind 等进行一个随机模拟的运行。通过查看消息序列图 MSC 能查看各个进程之间通过通道进行的消息交互(图 5)。

下面通过分析 leader 选举算法，来熟悉 iSpin 的使用过程，打开 Verification 标签，验证 p3 属性的选择如图 6 所示。

验证 p4 属性的如图 7 所示。

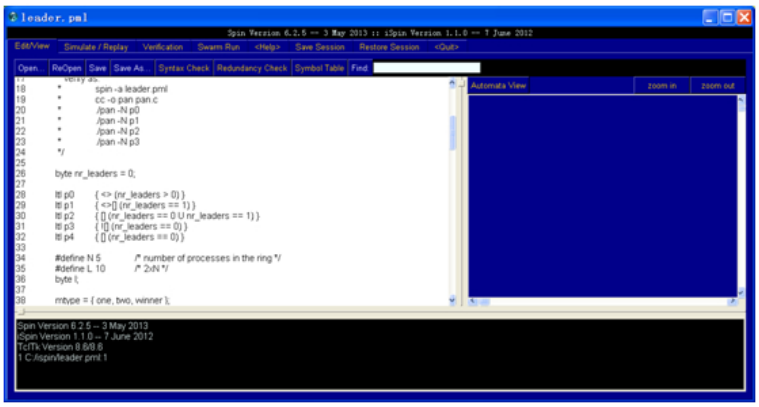

Figure 4. Edit/View label
图 4. Edit/View 标签

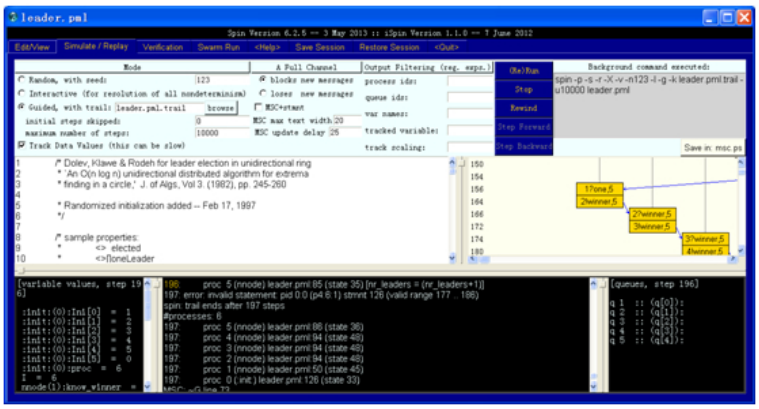

Figure 5. Simulate/Replay label
图 5. Simulate/Replay 标签

P4 属性在被验证出错之后，通过点击 Simulate/Replay 标签中的 Run 运行，能得到一条错误的路径，如图 8 所示。

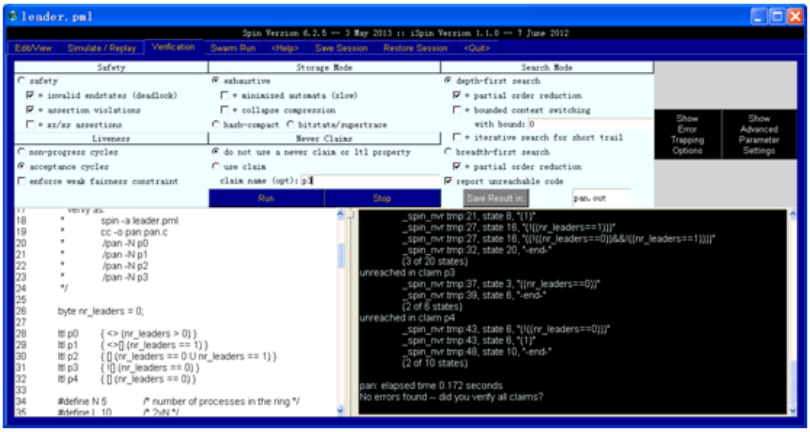

Figure 6. Verify a correct LTL property
图 6. 验证一个正确的 LTL 属性

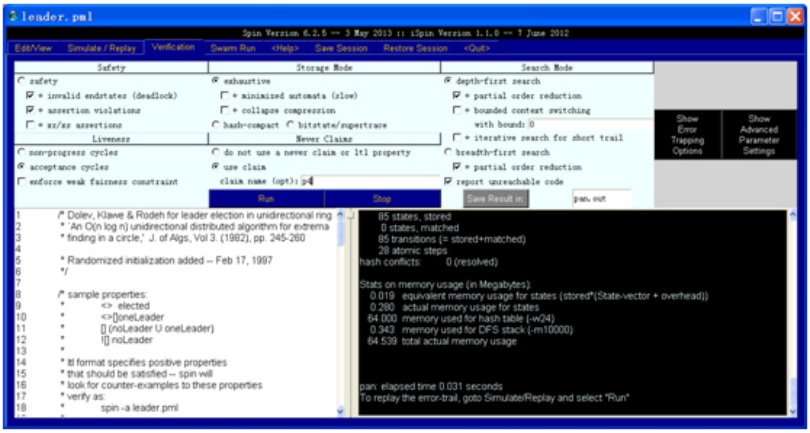

Figure 7. Verify an error LTL property
图 7. 验证一个错误的 LTL 属性

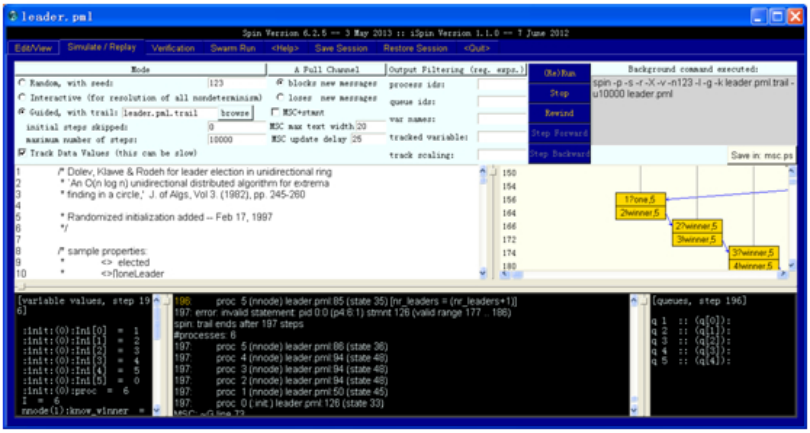

Figure 8. View a counterexample error
图 8. 查看出错的一个反例

我们选用了环选举算法，并验证了性质 p3：ltl p3{![] (nr_leaders == 0) }和性质 p4：ltl p4{ [] (nr_leaders == 0)}，并得出 p3：No errors found -- did you verify all claims?的结论，而 p4：To replay the error-trail, goto Simulate/Replay and select "Run"，在 Simulate/Replay 中我们找到了一条错误的路径。

7. 结语

SPIN 一个成功并被广泛接受的分析验证并发系统逻辑一致性的检测工具，其将先进的形式化验证方法应用在大型的软件系统中，对软件的有效性、可靠性进行分析。本文研究了 SPIN 的原理和结构，包括建模语言 Promela 和线性时序逻辑 LTL 到自动机的转换等。在此基础上，详细介绍了 SPIN 图形化界面工具 iSpin 的安装和使用，并通过具体实例进行了分析。模型检测可以有效的验证并发系统的正确性和其他性质，具有广阔的应用前景。

参考文献 (References)

[1] Holzmann, G.J. (2003) The SPIN model checker: Primer and reference manual. Addison-Wesley, Boston.

[2] Ben-Ari, M. (2008) Principles of the spin model checker. Springer Verlag, London.

[3] Spin Readme. http://spinroot.com/spin/Man/README.html

[4] Spin Sources. http://spinroot.com/spin/Src/index.html

[5] Spin verifier’s roadmap: Using iSpin. http://spinroot.com/spin/Man/GettingStarted.html

[6] Gerth, R., Peled, D., Vardi, M.Y. and Wolper, P. (1995) Simple on-the-fly automatic verification of linear temporal logic. *Proceedings of the* 15*th International Conference on Protocol Specification, Testing and Verification*, Warsaw, 3-18.

[7] Promela Manual Pages. http://spinroot.com/spin/Man/promela.html

[8] Boigelot, B. and Godefroid, P. (1996) Model checking in practice: An analysis of the ACCESS bus protocol using SPIN. 3*rd International Symposium of Formal Methods Europe Co-Sponsored by IFIP WG* 14.3 *Oxford*, 18-22 March 1996, 465-478.

[9] 孙守卿 (2006) 基于模型检测工具 SPIN 的安全协议分析和验证. 硕士学位论文, 兰州大学, 兰州.

[10] 刘俏威 (2008) SP 州模型检测的形式化分析机理研究及应用. 硕士学位论文, 南昌大学, 南昌.

[11] Raynal, M. (1988) Distributed algorithms and protocols. JohnWiley & Sons, New York.

[12] 易锦, 张文辉 (2006) 从基于迁移的扩展 Buhci 自动机到 Buchi 自动机. *软件学报*, **4**, 720-728.