# 基于 SPIN 的模块化模型检测方法研究

李兴锋[①] 张新常*[②③] 杨美红[②③] 阎保平[①]
[①](中国科学院计算机网络信息中心 北京 100190)
[②](山东省科学院计算中心 济南 250014)
[③](山东省计算机网络重点实验室 济南 250014)

**摘 要**：该文针对模型检测过程中所存在的状态爆炸问题，提出了一种基于模型检测工具 SPIN 的模块化模型检测方法。所提出的方法能够将指定的抽象模型分解成若干的模块，并对这些验证复杂度相对低的模块执行模型检测，以替代对原模型的模型检测。所提方法所用的分解过程保留了原模型所有的语义，同时不增加额外的语义，从而使得验证所有模块等同于验证原模型。理论和实验分析结果显示了所提方法的有效性。

**关键词**：模型检测；扩展有限状态自动机；状态爆炸
**中图分类号**：TP393 **文献标识码**：A **文章编号**：1009-5896(2011)-04-0902-06
**DOI**: 10.3724/SP.J.1146.2010.00751

# Study on Modularized Model Checking Method Based on SPIN

Li Xing-feng[①] Zhang Xin-chang[②③] Yang Mei-hong[②③] Yan Bao-ping[①]
[①](*Computer Network Information Center*, *Chinese Academy of Sciences*, *Beijing* 100190, *China*)
[②](*Shandong Computer Science Centre*, *Shandong Academy of Sciences*, *Jinan* 250014, *China*)
[③](*Shandong Provincial Key Laboratory of Computer Networks*, *Jinan* 250014, *China*)

**Abstract**: To address the state explosion problem in the procedure of model checking, this paper proposes a SPIN-based modularized model checking method. The proposed method divides the original abstraction model into some modules, and verifies all the divided modules, instead of verifying the original model. In the dividing process, the semantic of original model can be completely included in the modules, and no semantic is added in the process. Consequently, the original model passes the verification of model checking if and only if all the modules pass the verification. The theoretical and experimental results show that the proposed method is valid.

**Key words**: Model checking; Extended Finite State Machine (EFSM); State explosion

## 1 引言

模型检测是一种最早由 Clark 等人提出的形式化验证方法，它能够以较小的代价对系统进行有效验证。对于一些并发性和分布性明显的复杂系统，由于传统的验证技术(如测试)面临诸多问题，模型检测具有更加明显的优势。目前，最常用的模型检测技术绝大多数基于时态逻辑。根据时态逻辑类型的不同，可将时态逻辑模型检测分为基于线性时态逻辑的模型检测和基于计算树逻辑的模型检测。目前，模型检测技术已得到了广泛的研究[1–3]，并在全球范围内被应用到通信协议、并行系统和航空设计等领域。

典型的模型检测工具包括 UPPAAL，SPIN，NuSMV 和 Kronos 等。其中，SPIN 是一款典型的基于线性时态逻辑的模型检测工具。SPIN 用线性时态逻辑对系统属性进行描述，用 PROMELA 语言描述系统模型。由于扩展有限状态自动机 EFSM 模型具有良好的控制流和数据流描述能力，在 SPIN 中系统模型往往用 EFSM 模型加以描述。SPIN 具有结构简洁、自动化程度高等特点，从而备受相关行业和研究人员关注。近年内，也出现了一些基于 SPIN 的相关研究[4–7]。

如其它模型检测工具一样，SPIN 也面临状态爆炸问题，即在验证复杂系统时可达状态数量急剧增多而导致验证无法顺利进行。目前，解决模型检测状态爆炸问题的途径大体上可分为两个层次：在模型检测过程中缓解状态爆炸现象，如带界限的模型检测技术[3]和偏序搜索技术[8]等；通过对系统模型的

2010-07-15 收到，2010-11-02 改回
国家 973 计划项目(2009CB320502)，国家自然科学基金(61070039)，国家 863 计划项目(2009AA01Z145)和山东省科学院院博士基金(2010-12)资助课题
*通信作者：张新常 zhangxc@keylab.net

进一步抽象简化验证模型。由于模型检测是一种基于状态穷尽搜索的技术，前一种途径对某些复杂系统而言可能无法有效解决状态爆炸问题。抽象简化是一种解决状态爆炸问题的有力方法[9]。然而，在已有的文献中，抽象简化往往是作为一种方法论提出的，很少文献提出具体的、系统化的抽象简化方法。此外，对被验证模型进行整体简化往往浪费大量的时间和人力资源，且简化后的模型与简化前的模型难以保持完全一致。

针对上述问题，本文提出一种基于 SPIN 的模块化模型检测方法。所提出的方法能够将指定的抽象模型分解成若干等价的模块，且各模块具有相对低的验证复杂度，从而通过模块化验证的方法提供一种解决状态爆炸问题的有效途径。

## 2　一种基于 SPIN 的模块化模型检测方法

### 2.1 相关概念

下面将本文提出的模块化模型检测方法涉及到的一些概念进行介绍。

**定义 1**　设 $s_i$ 和 $s_j$ 是 EFSM 模型 $M$ 中的两个状态，若从状态 $s_i$ 到 $s_j$ 存在一条有向路径，则称从状态 $s_i$ 可以到达 $s_j$。

**定义 2**　给定 EFSM 模型 $M$ 中的某状态，若以其为尾状态的状态迁移数量为 1，则称该状态为单入度状态。若从 $M$ 中状态 $s_i$ 到状态 $s_j$ 仅存在一条路径，则称该路径为单入度路径。

在本文中，某状态的入度表示以该状态为尾状态(即状态迁移所进入的状态)的状态迁移数量。

**定义 3**　设 EFSM $M$ 中的终止状态集记为 $F$，一个分解终止状态集合 $D$ 为 $F$ 的一个子集，且满足(1)$D$ 中的任意一个状态不能到达其它状态；(2)不在 $D$ 中的终止状态必定能到达 $D$ 中至少一个状态。

进一步地，一个最大分解终止状态集合指包含元素数量最多的分解终止状态集合。特别地，当最大分解终止状态集合大小为 1 且存在不同的最大分解终止状态集合时，则此时最大分解终止状态集合特指由入度最大的终止状态所组成的最大分解终止状态集合。在本文中，EFSM 模型 $M$ 的最大分解终止状态集合记为 $D_{\max}(M)$。

**定义 4**　终止状态迁移为以某终止状态为尾状态的状态迁移。若一个终止状态迁移以终止状态 $e$ 为尾状态，则称该终止状态迁移与 $e$ 相关联。

在 EFSM 模型中，一个终止状态可能与多个终止状态迁移相关联。设某终止状态为 $e$，与 $e$ 相关联的所有终止状态迁移所组成的集合记为 $R(e)$。

### 2.2 模块分解

在 SPIN 中，描述 EFSM 模型的 PROMELA 程序被称为 PROMELA 模型。在本文所提出的模块化模型检测方法中，一个 EFSM 模型可能被等价地分解成若干的模块。在下文中，将称这些分解后的模块为子模型。下一节将介绍如何修改 PROMELA 模型表示分解后的子模型，本节将仅关注于 EFSM 模型的分解过程。算法 1(表 1)给出了 EFSM 模型 $M$ 分解成两个子模型的具体过程。

算法 1 中的模型 $M$ 应满足如下条件：(1)$|D_{\max}(M)|>1$；或(2)$|D_{\max}(M)|=1$ 且存在入度大于 1 的终止状态。算法 2 将通过改进原模型方式保证上述条件得以满足，详细情况可参见 2.3 节中相关描述。

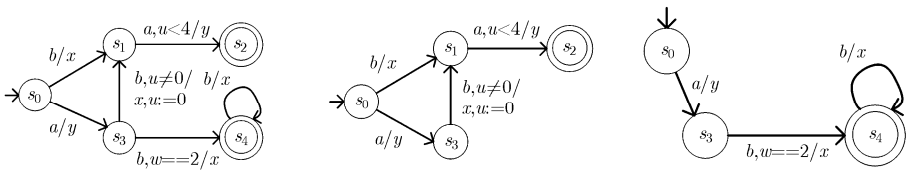
**表1 算法1：模型分解算法**

```
1  Procedure ModelPartition(M,D_max(M))
2  if (|D_max(M)|>1)
3      将 D_max(M)均分为两个集合，记为 P1 和 P2。
4  else
5      找到入度最大的终止状态，记为 e；
6      将 R(e)均分为两个集合，记为 P1 和 P2；
7  endif
8  for ∀ x(x∈P1) do
9      从 x 出发逆向遍历模型图 M，并在遍历过程中将 x 及经过的
       状态和状态迁移附加 sub1 标签；若 x 为状态迁移，则将 x 的
       尾状态也附加 sub1 标签；
10 endfor
11 for ∀ y(y∈P2) do
12     从 y 出发逆向遍历模型图 M，并在遍历过程中将 x 及经过的
       状态和状态迁移附加 sub2 标签；若 y 为状态迁移，则将 y 的
       尾状态也附加 sub2 标签；
13 endfor
14 所有带 sub1 标签的状态和状态迁移构成子模型 M_sub1；所有带
   sub2 标签的状态和状态迁移构成子模型 M_sub2。
```

在算法 1 第 3 和 6 行中，对集合 $A$ 进行均分的过程如下：根据该集合中元素的数量，将该集合分为两个集合 $A_1$ 和 $A_2$，使得 $A_1 \cap A_2 = \Phi$，且满足 $||A_1|-|A_2|| \leqslant 1$。在算法 1 第 9 和 12 行中，$P_1$ 和 $P_2$ 中的元素可能是状态或状态迁移(参见算法 1 中 2–7 行)。算法 1 中所述的逆向遍历等价于将 EFSM 模型图中的状态迁移方法逆转，并从指定状态或状态迁移开始按照某种搜索策略(深度优先搜索或宽度优先搜索)对改动后的图进行遍历。算法通过上述逆转搜索的方式找出所有与某终止状态或终止状态迁移有关的状态迁移路径，在将经过的路径附加子模型标签(sub1 或 sub2)，所有同一子模型标签组成的模型构成了子模型。设 $f$ 为 EFSM 模型 $M$ 中的终止状态数量，$n$ 为 $M$ 中的状态数量，$t$ 为 $M$ 中的状态迁移数量，EFSM 模型图以邻接表存储，则算法 1 的复杂度为 $O(f(n+m))$。更加具体的分析可参见本文特征

分析部分。

图 1 给出了一个模型分解实例。图 1(a) 给出了一个包含 5 个状态和 6 个状态迁移(即关联两个状态的有向边)的 EFSM 模型实例。其中，标签“$a,u\neq 0/x,u:=0$”表示当状态 $s_3$ 在输入符号 $a$ 且满足“$u\neq 0$”($u$ 为变量)时将进入状态 $s_1$，并且输出符号 $x$ 及执行“$u:=0$”操作。此外，带双圈的状态表示自动机的终止状态，即允许自动机在此状态下停止。图 1(b)，1(c)所示的两个子模型是通过算法 1 对图 1(a)所示的模型进行分解所得。在该实例中，最大分解终止




(a) EFSM 模型 (b) 子模型 $M_{\text{sub1}}$ (c) 子模型 $M_{\text{sub2}}$

图 1 EFSM 模型分解示例

状态集包含2个终止状态。

### 2.3 模块化模型检测方法

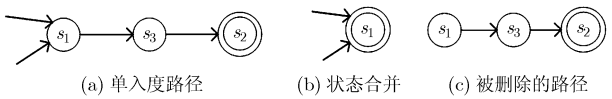
如文献[9]所述，抽象模型在多数情况下是在依据消息流建立的。在 SPIN 中，消息由 mtype 定义，消息交换通过消息信道完成。此外，PROMELA 语言是为描述系统模型而设计的，在描述模型方面具有比较简单的结构。因此，EFSM 模型和对应的 PROMELA 模型很容易相互转换。例如，PROMELA 模型中的“选项”(即“::”)语句可对应到一个 EFSM 模型状态，变量定义及初始值可对应到 EFSM 模型中的变量及其初始值。另外，PROMELA 模型中的消息流简洁直观，可比较容易地对应到 EFSM 模型中的路径。对 EFSM 模型分解后生成的子模型可在原 PROMELA 模型的基础上做相应修改后获得。一个比较简单的有效方法是将指定子模型不涉及的消息、消息信道、消息信道的某些操作和变量等删除，即可获得该子模型所对应的 PROMEAL 模型。本文将忽略更多的细节介绍。

算法 2(表 2)给出了所提模块化模型检测方法的具体过程。在算法 2 中，$T$ 是一个表示模型的变量。在上述算法中，$\lambda_{\max}(D_{\max}(T))=\max\{\lambda(x)|x\in F(T)\}$，其中，$\lambda(x)$表示状态 $x$ 的入度，$F(T)$表示模型 $T$ 的终止状态集合。在算法 2 第 5 行中，单入度路径替换过程 $re$(e)执行的前提是在从初始状态出发到终止状态 $e$ 的所有路径中，离 $e$ 最近的 $n$ 个状态(包括 $e$ 本身)的入度为 1。在上述 $n$ 个状态中，设离 $e$ 最远的状态为 $b$。在上述提前下，单入度路径替换过程 $re$(e)执行如下：将 $n$ 个状态中除状态 $e$ 和 $b$ 之外的状态及其相关状态迁移删除，并将 $e$ 和 $b$ 合并。在上述过程中，称从 $b$ 到 $e$ 之间的路径为被删除的路径。由于被删除的路径结构相对简单，可以手工验证或单独通过模型检测验证。事实上，这类路径在模型检测抽象过程中是应该尽量避免的路径[9]。

**表2 算法2：模块化模型检测算法**

| | |
|---|---|
| 1 | Procedure ModelCheck($M$) |
| 2 | $T\leftarrow M$; |
| 3 | 计算 $D_{\max}(T)$; |
| 4 | if $\lvert D_{\max}(\mathrm{T})\rvert=1$ 且 $\lambda_{\max}(D_{\max}(\mathrm{T}))=1$ then |
| 5 | 对 $D_{\max}(T)$中的唯一元素 $e$ 执行单入度路径替换过程 $re$(e)，并对被删除的路径进行单独验证； |
| 6 | endif |
| 7 | 执行 ModelPartition($M$, $D_{\max}(T)$)，若得到两个子模型 $M_{\text{sub1}}$ 和 $M_{\text{sub2}}$，则过程继续；否则，中断检测过程。 |
| 8 | 利用 SPIN 对 $M_{\text{sub1}}$ 进行模型检测，如果出现状态爆炸导致验证无法顺利进行，则：$T\leftarrow M_{\text{sub1}}$； Goto 2。 |
| 9 | 利用 SPIN 对 $M_{\text{sub2}}$ 进行模型检测，如果出现状态爆炸导致验证无法顺利进行，则：$T\leftarrow M_{\text{sub2}}$； Goto 2。 |

图 2 给出了一个单入度替换过程的实例。在该例中，图 2(a)中的单入度路径被替换为图 2(b)和 2(c)所示的结构。其中，图 2(c)部分已不再与原模型直接关联。




(a) 单入度路径 (b) 状态合并 (c) 被删除的路径

图 2 单入度路径替换实例

## 3 特征分析

下文分析中所提到的 EFSM 模型路径是从图的角度出发而言的。因此，本文合理地假定：(1)待验证的 EFSM 模型图中的节点都是可达的；(2)待验的 EFSM 模型是可终止的，即所有状态能够到达至少一个终止状态。若某 EFSM 模型不符合上述假定，说明该模型存在明显的设计缺陷，在其基础上进行

模型检测的必要性不大。

**引理 1**　设算法 1 的逆向遍历过程从状态 $s_i$ 开始，则逆向遍历所得到的子模型包括了在原模型 $M$ 中所有从初始状态 $s_0$ 出发到达 $s_i$ 的路径。

**证明**　不失一般性，设上述遍历过程得到的子模型为 $M_{\text{sub1}}$。假设在 $M$ 中存在一条从 $s_0$ 出发到达 $s_i$ 的路径 $p$，使得经过算法 1 得到的包含 $s_i$ 的子模型 $M_{\text{sub1}}$ 中不存在路径 $p$。设路径 $p$ 表示为状态迁移“$t_1, t_2,\cdots,t_k$”的级联。其中，$t_1$ 起始于状态 $s_0$，$t_k$ 终止于状态 $s_i$。由于 $p$ 不在 $M_{\text{sub1}}$ 中，“$t_1,t_2,\cdots,t_k$”中必然存在至少一个 $M$ 中的状态迁移不在 $M_{\text{sub1}}$ 中。进一步设离 $s_i$ 最近的不在 $M_{\text{sub}}$ 中的状态迁移为 $t_{k-j}$。现利用归纳法证明假设不成立：

(1)当 $j$=0 时，根据算法 1 容易得知所有以 $s_i$ 为尾状态的状态迁移都被 sub1 标记，故假设不成立；

(2)设 $j<m$ 时 $t_{k-j}$ 在 $M_{\text{sub}}$ 中，现证明 $t_{k-m}$ 也在 $M_{\text{sub1}}$ 中。由于 $t_{k-m+1}$ 在 $M_{\text{sub1}}$ 中，则说明 $t_{k-m+1}$ 的起始状态 (设为 $a$)已被 sub1 标记。则根据算法 1，以 $a$ 为尾状态的所有状态迁移都会被 sub1 标记，即 $t_{k-m}$ 也在 $M_{\text{sub1}}$ 中。从而证明该情况下假设不成立。

综上所述，该引理得证。

**定理 1**　给定一个 EFSM 模型 $M$，经过算法 1 分解得到的两个子模型包括了 $M$ 中的所有到达终止状态的路径。

**证明**　设 $M$ 的终止状态集合为 $F$。根据算法 1，分解 $M$ 存在如下两种情况：

(1)$|D_{\max}(M)|>1$；(2)$|D_{\max}(M)|=1$。

在情况(1)中，根据 $D_{\max}(M)$的定义可知，对于不在 $D_{\max}(M)$中的任意终止状态 $e$，$e$ 必定能到达 $D_{\max}(M)$中的某一状态。因此，所有终止状态必定在算法 1 逆向遍历中被标记。根据引理 1 可知，所有到达终止状态的路径都被包含在两个子模型中。

类似地，可以证明在情况(2)中所有到达终止状态的路径都被包含在两个子模型中。

综上所述，本定理得证。

**定理 2**　对任意一个 EFSM 模型 $M$ 中的终止状态 $e$，能够从初始状态 $s_0$ 出发到达 $e$ 的所有路径至少包含在一个子模型中。

**证明**　从定理 1 的证明过程可知，任意一个 EFSM 模型 $M$ 中的终止状态 $e$，必定在算法中被至少一个子模型标签(sub1 或 sub2)标记一次。一旦 $e$ 被标记，则根据逆向遍历过程，所有能到达它的路径都会被标记，即这些路径被包含在此时标记它的子模型中。因此，本定理得证。

**定理 3**　给定一个 EFSM 模型 $M$，其通过模型检测验证当且仅当其通过算法 2 所述的模型检测验证。

**证明**　在本定理中，不对模型检测过程所需的计算及存储等资源做限制。根据定理 1 和定理 2 可知，对某模型进行分解后所得到的两个子模型包含且仅包含原模型的路径信息(包括任意状态之间的关联信息)。因此，算法 1 所进行的分解操作从模型检测角度上来看是等价转换过程。在算法 2 中，可能需要对某模型进行单路径替换过程。经过替换的模型部分实际上是对模型的进一步抽象，而被删除的部分则通过其它方式或单独进行模型检测对其进行验证。因此，本定理得证。

根据定理 3 可以看出，本文所提出的模块化模型检测方法可以间接验证原模型，而不仅仅是抽象简化。

设 $nt(M)$表示一个 EFSM 模型 $M$ 中的状态迁移数量，则可以推导出定理 4。

**定理 4**　设算法 1 将一个 EFSM 模型 $M$ 分解成两个子模型，则对任意一个子模型(记为 $M_{\text{sub1}}$)满足：$nt(M_{\text{sub1}})<nt(M)$。

**证明**　根据算法 1，分解 $M$ 存在如下两种情况：(1)$|D_{\max}(M)|>1$；(2)$|D_{\max}(M)|=1$。

在情况(1)中，根据 $D_{\max}(M)$的定义可知，$D_{\max}(M)$中的任意两个状态不能存在一个状态到达另一个状态的情况。因此，根据算法 1 将划分后的两个子集 $P_1$ 和 $P_2$ 中，状态互相不能到达。进一步地，以 $P_1$ 中状态为尾状态的状态迁移必定不能经由 $P_1$ 中状态而到达 $P_2$ 中的状态，反之亦然。

在情况(2)中，算法 1 也得到分别由终止于同一状态的状态迁移所组成的两个子集 $P_1$ 和 $P_2$ 中。根据算法 1 中的逆向搜索法，$P_1$ 中的状态迁移必定不能在依据 $P_2$ 构造子模型的过程中被子模型标签标记，反之亦然。

综上所述，本定理得证。

设一个状态迁移 $t$ 起始于 $s$，终止于 $e$，从初始状态到 $s$ 共有 $n$ 条路径，从 $e$ 出发到达终止状态的所有路径中包括 $m$ 个状态迁移（不同路径中的状态迁移重复计数），则删除该状态迁移 $t$ 能减少对 $n(1+m)$个状态迁移的遍历。状态迁移数量是影响验证复杂度的主要因素之一。因此，定理 4 表明所提出的模块化模型检测方法能有效地减低单个子模型的验证复杂度。

## 4　实验分析

本部分实验是在文献[9]中所提到的两个系统模型的基础上进行的。其中，一个为常见的电话呼叫

系统模型(下文简称呼叫交换模型)，另一个为客户/服务器系统模型(下文简称 C/S 模型)。在呼叫交换模型中，存在两条消息流路径，分别为“摘机→拨号音→拨号→振铃音→挂机”和“摘机→拨号音→拨号→忙音→挂机”。C/S 模型包括请求，占用，拒绝，授权 和返回 5 类消息，包括一个服务器端和若干客户端。服务器端通过指定数量的代理提供服务，每个代理一次仅能为一个客户端提供服务。更多关于上述两个模型的信息可参见文献[9]中的 14.6 节和 15.3 节。本部分的实验环境包括：Intel Core 2 Duo 2.83 GHz；1.5 GB of RAM；Fedora Core 12。实验中所用的模型检测工具为 spin5.2.5。

在对呼叫交换模型的实验中，原 EFSM 模型(记为原模型)被分为两个子模型，分别记为子模型 1 和子模型 2。其中，子模型 1 完整地包含了“摘机→拨号音→拨号→振铃音→挂机”消息流，子模型 2 完整地包含了“摘机→拨号音→拨号→忙音→挂机”消息流。图 3 给出了模型化模型检测方法的效果图。在该图的横坐标中，“状态”表示模型检测过程中所到达的状态(注意：由于存在条件判定，一个 EFSM 模型中的状态可对应若干检测过程中的可达状态)，“匹配状态”表示在搜索状态空间过程中匹配的状态，“迁移”表示在搜索过程中所经历的状态迁移。通过该图可以看出，所提出的模块化模型检测方法能明显降低呼叫交换模型的验证复杂度。

在 C/S 模型的实验中，原 EFSM 模型(也被记为原模型)首先被分为两个子模型，记为子模型 1 和子模型 2。接下来将子模型 1 进一步分解成子模型 1.1 和子模型 1.2。其中，子模型 2 完整地包含经由占用消息而终止的数据流，子模型 1.1 完整地包含最后经由授权和返回消息终止的数据流，子模型 1.2 完整地包含了最后经由拒绝和返回消息终止的数据流。在下文中，用状态降低率、匹配状态降低率和状态迁移降低率 3 个指标来衡量模块化方法降低验证复杂度的程度。其中，状态降低率、匹配状态降低率和状态迁移降低率分别表示在子模型验证过程中的可达状态数量、匹配状态数量和经过的状态迁移数量在原模型验证过程中的相应数值的比值。图 4–图 6 给出了 C/S 模型检测过程中的验证复杂度情况。在上述图中，服务器端最多能同时提供一个代理，横坐标表示了客户端数量。在实验过程中，当客户端数量为 9 时，状态爆炸现象导致了对原模型的验证无法顺利进行，而所提方法则能在该情况下完成验证过程。通过实验结果可以看出，所提出的模块化模型检测方法能明显降低 C/S 验证的复杂度。此外通过图 4–图 6 还可以看出，随着客户端数量的增多(即系统复杂度增大)，模块化模型检测方法的效率会逐渐增强。

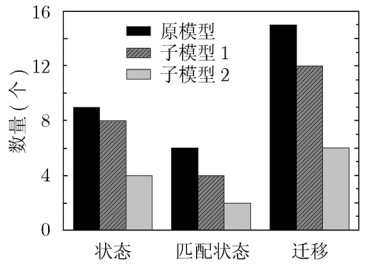



图 3 呼叫交换模型检测结果

## 5 结论

模型检测能够以较小的代价对系统模型进行有效验证，是一种重要的形式化验证方法。然而，在模型检测过程中，状态爆炸是需要克服的一个重要问题，且目前尚无理想的解决方法。针对上述问题，本文提出了一种基于模型检测工具 SPIN 的模块化模型检测方法。所提出的方法能够将指定的抽象模型分解成若干的独立的子模型，各子模型具有相对低的验证复杂度。在上述分解过程中，所有子模型共同包含了原模型的全部语义，即使得原模型通过模型验证当且仅当所有子模型通过模型验证。因此，所提出的模块化模型检测方法提供了一种解决状态爆炸问题的有效途径。

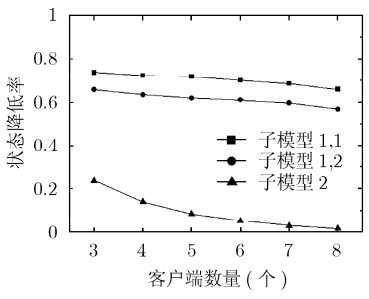



图 4 C/S 模型检测中的可达状态

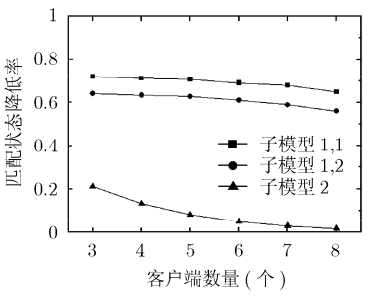



图 5 C/S 模型检测中的匹配状态

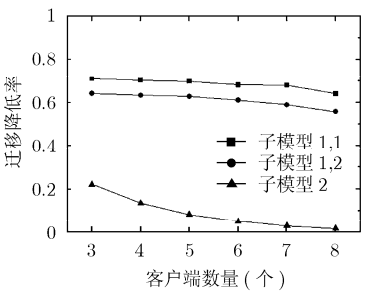



图 6 C/S 模型检测中的状态迁移

## 参 考 文 献


[1] Lu Si-mei, Zhang Jian-lin, and Luo Li-ming. The automatic verification and improvement of SET protocol model with SMV[C]. Proceedings of Information Engineering and Electronic Commerce, Ternopil, Ukraine, 2009: 433–436.

[2] McInnes A I. Model-checking the flooding time synchronization protocol control and automation[C]. Proceedings of ICCA 2009, Christchurch, 2009: 422–429.

[3] Biere A, Cimatti A, and Clarke E M, *et al*.. Bounded model checking[J]. *Advances in Computers*, 2003, 58: 117–148.

[4] Lima V, Talhi C, and Mouheb D, *et al*.. Formal verification and validation of UML 2.0 sequence diagrams using source and destination of messages[J]. *Electronic Notes in Theoretical Computer Science*, 2009, 254: 143–160.

[5] Li Jing and Li Jin-hua. Model checking the SET purchasing process protocol with SPIN[C]. Proceedings of 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, 2009: 1–4.

[6] Islam S M S, Sqalli M H, and Khan S. Modeling and formal verification of DHCP using SPIN[J]. *International Journal of Computer Science & Application*, 2006, 2(6): 145–159.

[7] O'Leary J, Saha B, and Tuttle M R. Model checking transactional memory with spin[C]. Proc. of the twenty-seventh ACM symposium on Principles of distributed computing, Montreal, QC, 2009: 335–342.

[8] Flanagan C and Godefroid P. Dynamic partial-order reduction for model checking software[J]. *ACM SIGPLAN Notices*, 2005, 40(1): 110–121.

[9] Holzmann G J. The Spin Model Checker: Primer and Reference Manual[M]. First edition, Boston: Addison Wesley, 2004: 217–360.



李兴锋：　男，1978 年生，博士，研究方向为下一代互联网络、软件测试等.

张新常：　男，1975 年生，博士，研究方向为软件测试、网络协议等.