马尔可夫决策过程的限界模型检测

周从华 邢支虎 刘志锋 王昌达
(江苏大学计算机科学与通信工程学院 江苏 镇江 212013)

摘 要 限界模型检测避免了符号模型检测反应式系统中构建二叉图时出现的空间快速增长,已经被证明是缓解状态空间爆炸问题的有力技术.文中遵循限界模型检测的思想,对马尔可夫决策过程提出一种限界模型检测技术,从而避免构建多端二叉图时空间的快速增长.具有非确定选择刻画能力是马尔可夫决策过程最大的特性,针对该特性首先定义概率计算树逻辑的限界语义,并证明其正确性;然后基于不同界下所计算概率度量序列的演化趋势,设计了限界检测过程终止的判断准则;最后将限界模型检测过程转换为线性方程组的求解问题.实验结果说明限界模型检测技术在证据较短的情况下,所需内存空间少于无界模型检测算法.

关键词 模型检测;限界模型检测;概率计算树逻辑;马尔可夫决策过程;状态空间爆炸
中图法分类号 TP301 **DOI 号** 10.3724/SP.J.1016.2013.02587

Bounded Model Checking for Markov Decision Processes

ZHOU Cong-Hua XING Zhi-Hu LIU Zhi-Feng WANG Chang-Da
(*School of Computer Science and Telecommunication Engineering, Jiangsu University, Zhenjiang, Jiangsu* 212013)

Abstract Bounded model checking has been proven to be a powerful technique for the verification of reactive systems, since it can avoid the space blow up of BDDs (Binary Decision Diagrams). To avoid the space blow up of MTBDDs (Multi Terminal Binary Decision Diagrams), a bounded model checking technique for Markov decision processes is proposed. The biggest feature of Markov decision processes is able to describe the nondeterministic choice. For this feature, the bounded semantics of the probabilistic computation tree logic is first presented, and its correctness is proven. Second, based on the evolution trend of the sequence consisting of probability measure calculated by the bounded model checking procedure, three termination criterions are given. Third, the bounded model checking procedure of the probabilistic computation tree logic is transformed into a linear equation. Finally, the experiment results show that compared with the unbounded model checking if the length of the witness is short then the bounded model checking needs less space.

Keywords model checking; bounded model checking; probabilistic computation tree logic; Markov decision processes; state space explosion

1 引 言

模型检测[1-2]是目前应用非常广泛的一种形式化验证技术,其最大特点是验证过程完全自动化.传统模型检测技术主要验证系统行为的绝对正确性,如两个进程不能同时访问临界区.然而很多随机系统关注的是某事件发生的概率,如 Ad hoc 网关注的

收稿日期:2012-08-02;最终修改稿收到日期:2013-10-31.本课题得到国家自然科学基金青年基金(61300228,61003288)、中德合作交流基金(6111130184)、江苏省自然科学基金(BK2010192)和教育部博士点基金(20093227110005)资助.**周从华**,男,1978 年生,博士,副教授,主要研究方向为模型检测、访问控制、模态逻辑.E-mail: chzhou@ujs.edu.cn.**邢支虎**,男,1988 年生,硕士研究生,主要研究方向为模型检测.**刘志锋**,男,1981 年生,博士,主要研究方向为形式化方法、模型检测.**王昌达**,男,1971 年生,博士,教授,主要研究领域为信息安全技术.

是不可靠信道上消息丢失的概率.传统模型检测中常用的时态逻辑,如计算树逻辑 CTL 和线性时态逻辑 LTL,无法刻画概率度量.为此研究人员提出了概率计算树逻辑 PCTL[3]等逻辑系统及相应的概率模型检测方法[4-5],其主要思想是在计算树和线性时态逻辑中引入概率度量算子,同时设计不同概率度量算子的检测方法.

在概率模型检测中,马尔可夫决策过程 MDP[4-5]是一种非常重要的随机系统模型.与传统模型检测一样,状态空间爆炸问题[6-7]是模型检测 MDP 实用化的主要瓶颈.一种克服该问题的自然的方法是将传统模型检测中的缓解空间爆炸的技术扩展应用到概率模型检测上.这些方法包括基于二叉决策图的状态空间压缩技术[8-9]、通过剥离与验证属性无关因素而合并状态空间的抽象技术[10]、对进程间交互次序进行等价类划分的偏序归约[11]与对称归约[12]技术、基于模块分析的组合推理[13]等.近年来研究人员从反驳证明的角度提出了一种全新的系统规模约简技术——限界模型检测[14-19],其主要特性是无需遍历全局空间,只需在验证属性必需的局部空间上即可完成验证过程.这种特性决定了将限界检测技术应用于模型检测 MDP 中,必能有效缓解存储空间溢出问题,本文的研究动机也源于此.

限界模型检测起源于 Biere 在 1999 年提出的基于命题公式可满足性判定的线性时态逻辑 LTL 的自动化验证过程.反驳证明是限界模型检测进行验证的主要原理,即通过逐步增加路径的长度来搜索使待证明性质失效的反例或者成立的证据.2002 年 Penczek 等人[20]提出了计算树逻辑 CTL 的限界模型检测方法.2003 年,Penczek 等人[21]面向多智体系统提出了时态认知逻辑 ACTLK 的限界模型检测方法.Lomuscio 等人[22]提出了实时认知逻辑 TECTLK 的限界模型检测算法.非概率逻辑系统的限界检测主要利用命题逻辑对反例或者证据的存在性进行编码,使得存在反例或者证据当且仅当对应的命题公式是可满足的.反例和证据的结构特征决定了命题逻辑有足够的能力对它们进行完全编码.而概率算子的反例和证据,其结构特征是状态转换关系满足一定的概率分布,命题逻辑对概率分布缺乏表达力.因此概率算子的特性决定了将限界检测技术扩展到 MDP 上,必会出现很多新的有价值的理论问题,这些新问题使得对概率算子的限界检测进行系统的研究成为必要.

MDP 是一种重要且常用的随机系统模型,本文结合 MDP 的特性,系统研究了 MDP 上概率计算树逻辑 PCTL 的限界检测问题,具体工作包括 3 个方面:

(1)在 MDP 上配置 PCTL 中不同算子的限界语义,并证明限界语义是对无界语义的逼近;

(2)设计了一套利用线性方程组刻画不同算子的限界语义的算法,使得线性方程组的解即为该算子对应的概率度量,从而 PCTL 限界语义的可满足性检测可通过判断线性方程组的解是否满足相应的概率度量约束即可;

(3)以实例说明在限制路径长度下计算的概率度量可能永远无法达到精确值,因此非概率逻辑限界检测中以设置完全界为检测终止性的准则已经失效,本文依据不同界下所计算概率度量序列的演化趋势,设计了新的限界检测过程终止的判断准则.

实验结果表明:

(1)限界模型检测得到的是一个非严格递增的概率度量序列,且与真实的概率度量误差越来越小;

(2)MDP 上 PCTL 的限界模型检测通过逐渐增加路径的长度来搜索属性为真的证据,因此对于具有较短的属性为真的证据的 MDP,能够快速地完成验证过程,并且在存储空间的需求上低于 PCTL 的无界模型检测算法.

文献[23]系统研究了离散时间马尔可夫链 DTMC 上的 PCTL 限界检测问题,包括 DTMC 上 PCTL 的限界语义、限界检测算法以及检测终止判定准则.文献[24]面向多智体系统研究了概率实时认知逻辑 PTCTLK 的限界检测.上述两篇论文的研究发现在概率算子的限界检测中路径长度作为检测过程终止的判别条件已经失效,必须设计新的判断准则.此外两篇论文均设计了基于线性方程组求解的限界模型检测算法,从而可以借助于数值计算工具完成检测过程.MDP 与 DTMC 最大不同在于 MDP 能够刻画非确定性选择,从而可以进行任务调度与优化分析.本文的工作是由于文献[23]的延生,重点解决非确定性选择给 PCTL 的限界语义以及检测算法带来的新的理论问题.表 1 详细比较了本文与文献[23-24]的工作.

表 1　相关研究工作的比较

论文	研究对象	对象特性	逻辑系统	技术框架	重点解决的问题
文献[23]	离散时间马尔可夫链	时间离散，概率转换关系	概率计算树逻辑 PCTL	限界检测：配置限界语义，设计限界检测算法，设计终止检测过程的规则	满足概率分布的转换关系的限界检测
文献[24]	概率实时解释系统	时间连续，概率转换关系	概率实时认知逻辑 PTCTLK	限界检测：配置限界语义，设计限界检测算法，设计终止检测过程的规则	实时，概率转换，多智体协作的限界检测
本文	马尔可夫决策过程	时间离散，概率转换关系，非确定性选择	概率计算树逻辑 PCTL	限界检测：配置限界语义，设计限界检测算法，设计终止检测过程的规则	在文献[23]的基础上解决非确定性选择的限界检测

本文第 2 节介绍与 MDP、PCTL 相关的基本概念；第 3 节探讨 MDP 限界模型检测的 3 个关键问题，包括 PCTL 限界语义的配置、限界语义可满足性的检测算法、终止检测过程的标准；第 4 节通过自稳定协议说明限界模型检测过程；第 5 节展示实验结果，并得出一些结论；第 6 节为提高概率度量计算的精度，探讨终止性判断准则的修正；第 7 节总结全文并展望未来的工作.

2　马尔可夫决策过程与概率计算树逻辑

PCTL 的语法和语义涉及概率度量的计算问题，因此首先回顾概率论的基本内容. 样本空间是由随机试验中所有可能发生的结果构成的集合，记为 Ω. Ω 的幂集 2^{Ω} 的子集 Π 称为 σ 代数，当且仅当满足下面 3 个条件：

$\Omega\in\Pi$；

如果 $E\in\Pi$，则 $\Omega\backslash E\in\Pi$；

如果 $E_1,E_2,\cdots\in\Pi$，则 $\bigcup_{i\geq 1}E_i\in\Pi$.

由样本空间 Ω、Ω 上的 σ 代数 Π 以及函数 $Prob:\Pi\to[0,1]$ 构成的三元组 $PS=(\Omega,\Pi,Prob)$ 称为概率空间，满足 $Prob(\Omega)=1$，且对 Π 中互不相交的序列 $E_1,E_2,\cdots$，$Prob(\bigcup_{i=1}^{\infty}E_i)=\bigcup_{i=1}^{\infty}Prob(E_i)$. 称 Π 中任何元素是可度量的.

如图 1 所示，MDP 是马尔可夫链的变种，其主要特点是允许概率和非确定性选择. 令 $Dist(S)$ 表示集合 S 上概率分布的集合，即满足 $\sum_{s\in S}\mu(s)=1$ 的

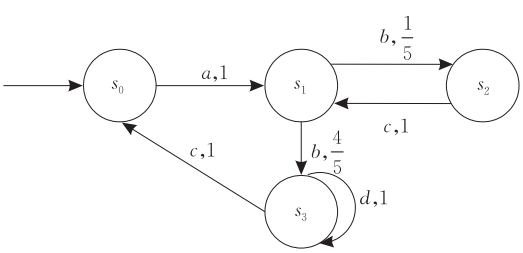

图 1　一个简单的 MDP M_1

函数 $\mu:S\to[0,1]$ 的集合. MDP 的形式化定义如下.

定义 1.　MDP $M=(S,s_{\text{in}},Act,Steps,Ap,L)$ 是一个六元组，这里

S 是有限状态集；

$s_{\text{in}}\in S$ 是初始状态；

Act 是动作集；

$Steps$：$S\to 2^{Act\times Dist(S)}$ 是转换概率函数，且满足对任意的 $(a,\mu)\in Steps(s)$，$\sum_{s'\in S}\mu(s')=1$；

Ap 是有限的原子命题集；

L：$S\to 2^{Ap}$ 是标记函数.

定义 2(路径).　设 M 为 MDP，M 中的路径 π 是具有下述形式的无穷序列：$s_0\xrightarrow{(a_1,\mu_1)}s_1\xrightarrow{(a_2,\mu_2)}s_2\cdots$，这里对任意的 $i\geq 0$，$s_i\in S$，$(a_{i+1},\mu_{i+1})\in Steps(s_i)$，$\mu_{i+1}(s_{i+1})>0$.

对于路径 π，引入记号 $\pi(i)$ 表示第 i 个状态. 对于有穷路径 π_{fin}，$last(\pi_{\text{fin}})$ 表示 π_{fin} 上的最后一个状态，$|\pi_{\text{fin}}|$ 表示路径的长度，即状态转换发生的次数. $Path^{\text{fin}}$，$Path$ 分别表示 M 中有穷和无穷路径集合，$Path_s^{\text{fin}}$，$Path_s$ 分别表示从 s 出发的有穷和无穷路径集.

定义 3.　令 M 为 MDP，映射 Adv：$Path^{\text{fin}}\to\{Steps(last(\pi_{\text{fin}}))\mid\pi_{\text{fin}}\in Path^{\text{fin}}\}$ 称为 M 上的一个调度.

例如对于图 1 中的 MDP M_1，存在这样的两个调度 Adv_1：$Adv_1(s_0)=(a,\mu_a)$，$Adv_1(s_0\xrightarrow{(a,\mu_a)}s_1)=(b,\mu_b)$，$Adv_1(s_0\xrightarrow{(a,\mu_a)}s_1\xrightarrow{(b,\mu_b)}s_3)=(c,\mu_c)$，$\cdots$；$Adv_2$：$Adv_1(s_0)=(a,\mu_a)$，$Adv_1(s_0\xrightarrow{(a,\mu_a)}s_1)=(b,\mu_b)$，$Adv_1(s_0\xrightarrow{(a,\mu_a)}s_1\xrightarrow{(b,\mu_b)}s_3)=(d,\mu_d)$，$\cdots$.

引入记号 Adv_M 表示 M 上所有调度的集合，记号 $Path_s^{Adv}$ 表示从 s 出发和 Adv 对应的路径集. 对于 MDP M、状态 s 和调度 Adv，令 $\Omega=Path_s^{Adv}$ 为 M 中从 s 出发和 Adv 对应的路径集. Π 是 σ 代数，定义为 $\Pi=\{C(\rho)\mid\rho$ 是 Ω 中某条路径的前缀$\}$，这里 $C(\rho)=$

$\{\pi\in\Omega\mid\rho$ 是 π 的有限前缀$\}$. Π 上的概率度量 $Prob_s^{Adv}$ 定义为 $Prob_s^{Adv}(C(s\xrightarrow{(a_1,\mu_1)}s_2\xrightarrow{(a_2,\mu_2)}s_3\cdots s_n))=\prod_{2\le i\le n}\mu_{i-1}(s_i)$. 这样我们从 MDP M、状态 s 和调度 Adv 演绎出了一个概率空间.

定义 4(概率计算树逻辑 PCTL). PCTL 由状态公式和路径公式构成. 给定原子命题集 Ap,PCTL 状态公式递归定义如下:$\phi::=\text{true}\mid a\mid\phi\wedge\phi\mid\phi\vee\phi\mid\neg\phi\mid P_{\infty p}(\psi)$,这里 $a\in Ap,\infty\in\{\le,<,\ge,>\}$,$p\in[0,1]$,$\psi$ 是一条路径公式. PCTL 路径公式递归定义如下:$\psi::=\text{X}\phi\mid\text{F}\phi\mid\text{G}\phi\mid\phi\text{U}\phi\mid\phi\text{R}\phi$,这里 ϕ 是状态公式.

MDP 上 PCTL 的满足性关系定义如下.

定义 5(概率计算树逻辑 PCTL 的满足性关系). 令 $a\in Ap$ 为原子命题,$M=(S,s_{\text{in}},Act,Steps,Ap,L)$ 是 MDP,$s\in S$,ϕ_1,ϕ_2 是 PCTL 状态公式,ψ 是 PCTL 路径公式. 对于状态公式满足性关系 $\models$ 定义为

$s\models a$ 当且仅当 $a\in L(s)$;

$s\models\neg\phi_1$ 当且仅当 $s\not\models\phi_1$;

$s\models\phi_1\wedge\phi_2$ 当且仅当 $s\models\phi_1$ 且 $s\models\phi_2$;

$s\models\phi_1\vee\phi_2$ 当且仅当 $s\models\phi_1$ 或者 $s\models\phi_2$;

$s\models P_{\infty p}(\psi)$ 当且仅当对所有的 $Adv\in Adv_M$,$Pr(s,\psi,Adv)\infty p$,这里 $Pr(s,\psi,Adv)=Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models\psi\})$.

对于 M 中的路径 π,满足性关系 $\models$ 定义为

$\pi\models\text{X}\phi_1$ 当且仅当 $\pi(1)\models\phi_1$;

$\pi\models\text{F}\phi_1$ 当且仅当存在自然数 i 使得 $\pi(i)\models\phi_1$;

$\pi\models\text{G}\phi_1$ 当且仅当对任意的自然数 i,$\pi(i)\models\phi_1$;

$\pi\models\phi_1\text{U}\phi_2$ 当且仅当存在自然数 j 使得 $\pi(j)\models\phi_2$,且对任意小于 j 的自然数 i,$\pi(i)\models\phi_1$;

$\pi\models\phi_1\text{R}\phi_2$ 当且仅当对任意的自然数 j,$\pi(j)\models\phi_2$,或者存在自然数 k 使得 $\pi(k)\models\phi_1$,且对任意不大于 k 的自然数 i,$\pi(i)\models\phi_2$.

3 PCTL 的限界模型检测

3.1 PCTL 的等价性

限界语义的定义必须确保限界语义可满足时,属性在局部和全局空间中同时成立. PCTL 中不涉及概率度量的算子,即 X,F,G,U,R,其限界语义等同于在 LTL 限界检测中的限界语义. 在限界检测过程中满足约束条件的路径会随着界的增长而增加,即概率度量会逐渐增加,这使得限界检测可直接应用于算子 $P_{\ge p}$ 和 $P_{>p}$,而不适用于算子 $P_{\le p}$ 和 $P_{<p}$ 类型的算子. 因此为了实现对整个 PCTL 公式的限界检测,本小节探讨能否在保持 PCTL 表达力的情况下,将概率约束限制为 $\ge p$ 或者 $>p$ 的形式.

定义 6(PCTL 公式的等价). 令 ϕ,φ 为任意的 PCTL 状态公式,如果对任一 MDP M,$s\in S$,$s\models\phi$ 当且仅当 $s\models\varphi$,则称 ϕ、φ 是等价的,记为 $\phi\equiv\varphi$.

依据定义 5 描述的 PCTL 的无界语义,可直接得出如下等价关系:

$P_{\le p}(\text{X}\phi)\equiv P_{\ge 1-p}(\text{X}\neg\phi)$;$P_{<p}(\text{X}\phi)\equiv P_{>1-p}(\text{X}\neg\phi)$;

$P_{\le p}(\text{F}\phi)\equiv P_{\ge 1-p}(\text{G}\neg\phi)$;$P_{<p}(\text{F}\phi)\equiv P_{>1-p}(\text{G}\neg\phi)$;

$P_{\le p}(\text{G}\phi)\equiv P_{\ge 1-p}(\text{F}\neg\phi)$;$P_{<p}(\text{G}\phi)\equiv P_{>1-p}(\text{F}\neg\phi)$;

$P_{\le p}(\phi\text{U}\varphi)\equiv P_{\ge 1-p}(\neg(\phi\text{U}\varphi))\equiv P_{\ge 1-p}(\neg\phi\text{R}\neg\varphi)$;

$P_{<p}(\phi\text{U}\varphi)\equiv P_{>1-p}(\neg(\phi\text{U}\varphi))\equiv P_{>1-p}(\neg\phi\text{R}\neg\varphi)$;

$P_{\le p}(\phi\text{R}\varphi)\equiv P_{\ge 1-p}(\neg(\phi\text{R}\varphi))\equiv P_{\ge 1-p}(\neg\phi\text{U}\neg\varphi)$;

$P_{<p}(\phi\text{R}\varphi)\equiv P_{>1-p}(\neg(\phi\text{R}\varphi))\equiv P_{>1-p}(\neg\phi\text{U}\neg\varphi)$.

上面的等价关系说明可将 $\le(<)p$ 的概率约束转换为 $\ge(>)p$ 的约束. 现在考察 $s\models\neg P_{\le p}(\psi)$,即 $s\not\models P_{\le p}(\psi)$,这意味着存在一个调度 $Adv\in Adv_M$ 使得 $Pr(s,\psi,Adv)>p$. 为了尽可能地保持 PCTL 表达力,在 PCTL 中引入公式 $P_{\infty p}^{\exists}(\psi)$,其语义为 $s\models P_{\infty p}^{\exists}(\psi)$ 当且仅当存在一个 $Adv\in Adv_M$ 使得 $Pr(s,\psi,Adv)\infty p$. 限界模型检测的基本原理是通过限制路径长度从而在有限空间中逐步搜索可用于反驳的证据或者反例,因此限界检测不适用于算子 $P_{<p}^{\exists}(\psi)$ 和 $P_{\le p}^{\exists}(\psi)$.

在保证 PCTL 表达力的前提下,现在定义 PCTL 的子集 $\text{PCTL}_{\ge}^{\exists}$,从而实现 PCTL 的限界模型检测. 形式化地讲,原子命题集 Ap 上的 $\text{PCTL}_{\ge}^{\exists}$ 状态公式定义如下:$\phi::=\text{true}\mid a\mid\neg a\mid\phi\wedge\phi\mid\phi\vee\phi\mid P_{\infty p}(\psi)\mid P_{\infty p}^{\exists}(\psi)$,这里 $a\in Ap,\infty\in\{\ge,>\}$,$p\in[0,1]$,$\psi$ 是一条路径公式. $\text{PCTL}_{\ge}^{\exists}$ 路径公式定义如下:$\psi::=\text{X}\phi\mid\text{F}\phi\mid\text{G}\phi\mid\phi\text{U}\phi\mid\phi\text{R}\phi$,这里 ϕ 是状态公式. $\text{PCTL}_{\ge}^{\exists}$ 是由上述状态公式和路径公式组成的逻辑系统.

3.2 PCTL 的限界语义

定义 7($\text{PCTL}_{\ge}^{\exists}$ 的限界语义). 令 $M=(S,s_{\text{in}},Act,Steps,Ap,L)$ 为 MDP,$a\in Ap$ 为原子命题,$s\in S$,k 为自然数(称为界),ψ 是 $\text{PCTL}_{\ge}^{\exists}$ 路径公式,ϕ_1、ϕ_2 是 $\text{PCTL}_{\ge}^{\exists}$ 状态公式. 状态公式的满足性关系 $\models_k$ 定义为

$s\models_k a$ 当且仅当 $a\in L(s)$;$s\models_k\neg a$ 当且仅当 $a\notin L(s)$;

$s\models_k\phi_1\wedge\phi_2$ 当且仅当 $s\models_k\phi_1$ 且 $s\models_k\phi_2$;

$s\models_k\phi_1\vee\phi_2$ 当且仅当 $s\models_k\phi_1$ 或者 $s\models_k\phi_2$；

$s\models_k P_{\infty p}(\phi)$ 当且仅当对所有的 $Adv\in Adv_M$，$Pr(s,\psi,Adv,k)\infty p$，这里 $Pr(s,\psi,Adv,k)=Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$；

$s\models_k P^{\exists}_{\infty p}(\phi)$ 当且仅当存在某个 $Adv\in Adv_M$ 使得 $Pr(s,\psi,Adv,k)\infty p$.

对于 M 中的路径 π，满足性关系 $\models_k$ 定义为

$\pi\models_k X\phi_1$ 当且仅当 $k\geq 1$ 且 $\pi(1)\models_k\phi_1$；

$\pi\models_k F\phi_1$ 当且仅当存在自然数 $i\leq k$ 使得 $\pi(i)\models_k\phi_1$；

$\pi\models_k G\phi_1$ 当且仅当对任意的自然数 $i\leq k$，$\pi(i)\models_k\phi_1$，且对任意的自然数 $j>k$，存在自然数 $h\leq k$ 使得 $\pi(j)=\pi(h)$；

$\pi\models_k\phi_1 U\phi_2$ 当且仅当存在自然数 $j\leq k$ 使得 $\pi(j)\models_k\phi_2$，且对任意的小于 j 的自然数 i，$\pi(i)\models_k\phi_1$；

$\pi\models_k\phi_1 R\phi_2$ 当且仅当：(1) 对任意的自然数 $i\leq k$，$\pi(i)\models_k\phi_2$，且对任意的自然数 $j>k$，存在自然数 $h\leq k$ 满足 $\pi(j)=\pi(h)$；或者 (2) 存在自然数 $m\leq k$ 满足 $\pi(m)\models_k\phi_1$，且对任意的自然数 i，如果 $i\leq m$，则 $\pi(i)\models_k\phi_2$.

现在考察定义 7 中 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$ 的可度量性. 对于算子 X，$\{\pi\in Path_s^{Adv}\mid\pi\models_k X\phi_1\}$ 是所有 $C(s_0\xrightarrow{(a_1,\mu_1)}s_1)$ 的并集，其中 $s_0=s$，$s_1\models_k\phi_1$. 对于算子 F，$\{\pi\in Path_s^{Adv}\mid\pi\models_k F\phi_1\}$ 是所有 $C(s_0\xrightarrow{(a_1,\mu_1)}s_1\xrightarrow{(a_2,\mu_2)}\cdots s_i)$ 的并集，其中 $i\leq k$，$s_0=s,s_i\models_k\phi_1$. 对于 U 算子，$\{\pi\in Path_s^{Adv}\mid\pi\models_k\phi_1 U\phi_2\}$ 是所有 $C(s_0\xrightarrow{(a_1,\mu_1)}s_1\xrightarrow{(a_2,\mu_2)}\cdots s_i)$ 的并集，其中 $i\leq k,s_0=s,s_0\models_k\phi_1,\cdots,s_{i-1}\models_k\phi_1,s_i\models_k\phi_2$. 由 σ 代数的定义可知，可度量集合的并仍是可度量的，因此对于 X、F、U 算子，$Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$ 是可度量的. 可度量性说明可以直接利用 $Prob_s^{Adv}(C(s\xrightarrow{(a_1,\mu_1)}s_2\xrightarrow{(a_2,\mu_2)}s_3\cdots s_k))$ 计算出 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$ 的值.

现在考察 G 算子. 对于 X、F、U 算子，$Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$ 是可度量的，本质在于如果有穷路径满足 ψ，则以该有穷路径为前缀的所有路径都满足 ψ，因此只需计算有穷路径发生的概率即可得 $Prob_s^{Adv}$ 的值. 而对于 G 算子，有穷路径不能反映以该路径为前缀的无穷路径的情况，因此 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k G\phi_1\})$ 难以直接度量. 在 3.4 节的限界检测算法部分我们将采用下近似的方法逼近 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k G\phi_1\})$. R 算子可以分解为 G 与 U 算子考虑，这里不再赘述.

为了判断一个状态是否满足 $P_{\infty p}(\phi)$，$P^{\exists}_{\infty p}(\phi)$，理论上必须计算出任一调度 Adv 下 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})$ 的值，然后再计算出最大值和最小值，并通过与 p 比较得出公式的真假. 实际上在 3.4 节的限界检测算法部分，我们设计了一种递归的方式可直接计算出最大值和最小值，从而避免了计算任意调度下 $Prob_s^{Adv}$ 的值.

定理 1.　令 $M=(S,s_{in},Act,Steps,Ap,L)$ 为 MDP，$s\in S$，$a\in Ap$ 为原子命题，k 为自然数，ϕ 是 $PCTL^{\exists}_{\geq}$ 状态公式. 如果 $s\models_k\phi$，则 $s\models\phi$.

证明.　证明过程通过对 ϕ 的长度实施归纳来完成.

情况 1. $\phi=a$.

$s\models_k a$ 说明 $a\in L(s)$，依据定义 5 中描述的相应的满足性关系，直接可得 $s\models a$.

情况 2. $\phi=\neg a$.

$s\models_k\neg a$ 说明 $a\notin L(s)$，依据定义 5 中描述的相应的满足性关系，直接可得 $s\not\models a$.

情况 3. $\phi=\phi_1\wedge\phi_2$.

$s\models_k\phi_1\wedge\phi_2$ 说明 $s\models_k\phi_1$，$s\models_k\phi_2$. 由归纳假设可知 $s\models\phi_1$，$s\models\phi_2$，即 $s\models\phi_1\wedge\phi_2$.

情况 4. $\phi=\phi_1\vee\phi_2$.

$s\models_k\phi_1\vee\phi_2$ 说明 $s\models_k\phi_1$ 或者 $s\models_k\phi_2$. 由归纳假设可知 $s\models\phi_1$ 或者 $s\models\phi_2$，即 $s\models\phi_1\vee\phi_2$.

情况 5. $\phi=P_{\geq p}(\psi)$.

$s\models_k P_{\geq p}(\psi)$ 说明对任意的调度 $Adv\in Adv_M$，$Pr(s,\psi,Adv,k)\geq p$，即 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})\geq p$. X、F、U 3 个时态算子的限界语义与 LTL 限界检测技术中的定义一致，因此有 $\pi\models_k\psi$ 蕴含 $\pi\models\psi$，从而 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models\psi\})\geq Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})\geq p$，即 $s\models P_{\geq p}(\psi)$. 对于 G 算子，由 $\pi\models_k G\phi_1$ 的定义可知对任意的自然数 $i\leq k$，$\pi(i)\models_k\phi_1$，对任意的自然数 $j>k$，必存在自然数 $h\leq k$ 使得 $\pi(j)=\pi(h)$. ϕ_1 是状态公式，因此 $\pi(h)\models_k\phi_1$ 蕴含了 $\pi(j)\models_k\phi_1$. 因此对任意的自然数 $l\geq 0$，$\pi(l)\models_k\phi_1$. 由归纳假设可知 $\pi(l)\models\phi_1$，因此 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models G\phi\})\geq Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k G\phi\})\geq p$，即 $s\models P_{\geq p}(\psi)$. 对于算子 R，$\phi_1 R\phi_2\equiv G\phi_2\vee(\phi_2 U(\phi_1\wedge\phi_2))$，因此算子 R 的证明可以归结为算子 G 和 U 的证明.

情况 6. $\phi=P_{>p}(\psi)$.

证明类似于情况 5.

情况 7. $\phi=P^{\exists}_{\geq p}(\psi)$.

$s\models_k P^{\exists}_{\geq p}(\psi)$说明存在一个调度 $Adv\in Adv_M$使得 $Pr(s,\psi,Adv,k)\geq p$，即 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid \pi\models_k\psi\})\geq p$. 对于时态算子 X、F、G、U，其限界语义与文献[14]中的定义一致，因此有 $\pi\models_k\psi$ 蕴含 $\pi\models\psi$，从而 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models\psi\})\geq Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k\psi\})\geq p$，即$s\models P_{\geq p}(\psi)$. 对于 G 算子，由 $\pi\models_k G\phi_1$的定义可知对任意的自然数 $i\leq k$，$\pi(i)\models_k\phi_1$，对任意的自然数 $j>k$，必存在自然数 $h\leq k$ 使得 $\pi(j)=\pi(h)$. ϕ_1 是状态公式，因此 $\pi(h)\models_k\phi_1$ 蕴含了 $\pi(j)\models_k\phi_1$. 因此对任意的自然数 $l\geq 0$，$\pi(l)\models_k\phi_1$. 由归纳假设可知 $\pi(l)\models\phi_1$，因此 $Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models G\phi\})\geq Prob_s^{Adv}(\{\pi\in Path_s^{Adv}\mid\pi\models_k G\phi\})\geq p$，即 $s\models P_{\geq p}(\psi)$. 对于算子 R，$\phi_1 R\phi_2\equiv G\phi_2\vee(\phi_2 U(\phi_1\wedge\phi_2))$，因此算子 R 的证明可以归结为算子 G 和 U 的证明.

情况 8. $\phi=P^{\exists}_{\geq p}(\psi)$.

证明类似于情况 7. 证毕.

定理 1 保证了限界语义定义的正确性，进而使得我们可以通过逐步增加路径长度的方式，得到精确概率度量的下近似.

3.3 限界模型检测过程终止的判断

定理 1 表明限界语义是对无界语义的逼近，即如果存在自然数 k 使得 $s\models_k\phi$，则断言 $s\models\phi$. 现在的问题是当 $s\not\models_k\phi$ 时，在增加 k 的值继续搜索还是停止搜索两者之间需要作出正确的选择. 如果继续搜索，不可能对界 k 无限增加下去，必须设计一套终止约束条件；如果停止搜索可能造成 $s\not\models\phi$ 的假象. 因此必须设计一个准则来决定 $s\not\models_k\phi$ 时的下一步选择.

定义 8. 称自然数 CT 是完全界当且仅当如果 $s\models\phi$，则一定存在自然数 $k\leq CT$ 使得 $s\models_k\phi$.

CTL 和 LTL 的限界模型检测因为不涉及极限运算，所以完全界一定存在，从而在知道 CT 的情况下，当 $s\not\models_k\phi$ 时如果 k 超过完全界，则停止搜索，并返回信息：$s\not\models\phi$，否则继续增加 k 的值进行搜索. 但是对于 $PCTL^{\exists}_{\geq}$ 限界模型检测，完全界则不一定存在.

考察图 1 所示的 MDP M_1. 对于状态 s，引入原子命题 a_s 表示当前状态为 s. 考察属性 $P_{=1}(Fa_{s_3})$. 通过计算发现，对任意的调度 Adv，$Pr(s_0,Fa_{s_3},Adv)=Prob_{s_0}^{Adv}\{\pi\in Paths_{s_0}\mid\pi\models Fa_{s_3}\}=Prob_{s_0}^{Adv}\{\pi=s_0,s_1,(s_2,s_1)^r,s_3,\cdots\mid r\geq 0\}=\frac{4}{5}+\frac{1}{5}\cdot\frac{4}{5}+\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{4}{5}+\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{4}{5}+\cdots=1$，即 $s_0\models P_{=1}(Fa_{s_3})$. 令 k 为界，则 $Pr(s_0,Fa_{s_3},Adv,k)=Prob_{s_0}^{Adv}\{\pi\in Paths_{s_0}\mid\pi\models_k Fa_{s_3}\}=Prob_{s_0}^{Adv}\left\{\pi=s_0,s_1,(s_2,s_1)^r,s_3,\cdots\mid r\leq\frac{k}{2}-1\right\}=\frac{4}{5}+\frac{1}{5}\cdot\frac{4}{5}+\frac{1}{5}\cdot\frac{1}{5}\cdot\frac{4}{5}+\cdots+\left(\frac{1}{5}\right)^{\left[\frac{k}{2}-1\right]}\cdot\frac{4}{5}=1-\left(\frac{1}{5}\right)^{\left[\frac{k}{2}-1\right]+1}$.

比较 $Pr(s_0,Fa_{s_3},Adv)$和 $Pr(s_0,Fa_{s_3},Adv,k)$，可以发现对于任意的有限界 k，$Pr(s_0,Fa_{s_3},Adv,k)<Pr(s_0,Fa_{s_3},Adv)$. 换句话，对于属性 $P_{=1}(Fa_{s_3})$，尽管 $s_0\models P_{=1}(Fa_{s_3})$，但不存在一个有限的自然数 k，使得 $s_0\models_k P_{=1}(Fa_{s_3})$.

现在探讨完全界存在的条件.

引理 1. 令 $M=(S,s_{in},Act,Steps,Ap,L)$是 MDP，$\phi$ 是 $PCTL^{\exists}_{\geq}$ 状态公式，ψ 是 $PCTL^{\exists}_{\geq}$ 路径公式. 则对于任意的自然数 k，属性在 k 步空间中成立，在 $k+1$ 步空间中也成立，即$s\models_k\phi\rightarrow s\models_{k+1}\phi$，$\pi\models_k\psi\rightarrow\pi\models_{k+1}\psi$.

通过对 $PCTL^{\exists}_{\geq}$ 公式的长度进行归纳即可完成引理 1 的证明，故这里不给出其证明过程.

定理 2. 令 $M=(S,s_{in},Act,Steps,Ap,L)$为 MDP，$\psi$ 是 $PCTL^{\exists}_{\geq}$ 路径公式，$p=\min\{Pr(s_{in},\psi,Adv)\mid Adv\in Adv_M\}$. 则对于公式 $P_{\geq r}(\psi)(r<p)$，一定存在完全界 CT 使得 $s_{in}\models P_{\geq r}(\psi)$ 时 $s_{in}\models_{CT}P_{\geq r}(\psi)$.

证明. 由引理 1 可知，$Pr(s_{in},\psi,Adv,k)$随着 k 的增加而不断增加，且 $\lim_{k\to\infty}Pr(s_{in},\psi,Adv,k)=Pr(s_{in},\psi,Adv)$. 令 $\xi=p-r$，由极限的定义可知存在自然数 k_ξ，当 $k>k_\xi$时 $|Pr(s_{in},\psi,Adv,k)-Pr(s_{in},\psi,Adv)|<\xi$. 此时取 $CT=k_\xi+1$，则 $|Pr(s_{in},\psi,Adv,CT)-Pr(s_{in},\psi,Adv)|<\xi$，即 $Pr(s_{in}\models_{CT}\psi)>p-\xi$，从而 $s_{in}\models_{CT}P_{\geq r}(\psi)$. 证毕.

定理 2 说明在何种条件下 MDP 上的完全界是存在的. 但是这种存在性条件依赖于事先计算 $Pr(s_{in},\psi,Adv)$，因此不具有实用性. 不存在完全界使得我们无法以设置路径长度的上限来决定检测过程何时终止，因此需要提出新的判别标准. 我们拟通过刻画不同界下所计算概率度量序列的演化趋势来执行判断.

回顾一下，判断 s 是否满足 $P_{\propto p}(\psi)$关键在于对所有 $Adv\in Adv_M$，$Pr(s,\psi,Adv)\propto p$. 如果对所有的 $Adv\in Adv_M$，$Pr(s,\psi,Adv)$的某个下近似满足$\propto p$，则 $Pr(s,\psi,Adv)\propto p$. 因此我们计算 $\min\{Pr(s_{in},\psi,Adv)\mid Adv\in Adv_M\}$的下近似 $\min\{Pr(s_{in},\psi,Adv,$

$k)\mid Adv\in Adv_M\}(k\geq 0)$. 同理，对于 $P^{\exists}_{\propto p}(\psi)$ 计算 $\max\{Pr(s_{in},\psi,Adv)\mid Adv\in Adv_M\}$ 的下近似 $\max\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}(k\geq 0)$. 我们以 $P_{\propto p}(\psi)$ 为例说明 $PCTL^{\exists}_{\geq}$ 限界模型检测过程的终止性判断准则.

判断准则 1.　$PCTL^{\exists}_{\geq}$ 限界模型检测的终止性判断.

输入：MDP $M=(S,s_{in},Act,Steps,Ap,L)$，$PCTL^{\exists}_{\geq}$ 路径公式 ψ，预先设置的终止标准 ξ，预先设置的计算步长 m

输出：$\min\{Pr(s_{in},\psi,Adv)\mid Adv\in Adv_M\}$ 的下近似值

1. 令 $k=1$，计算 $\min\{Pr(s_{in},\psi,Adv,0)\mid Adv\in Adv_M\}$，$\min\{Pr(s_{in},\psi,Adv,1)\mid Adv\in Adv_M\}$；

2. While $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}-\min\{Pr(s_{in},\psi,Adv,k-1)\mid Adv\in Adv_M\}\geq\xi\wedge k\leq m$ do {令 $k=k+1$，计算 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$；}

3. 输出 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$.

引理 1 告诉我们，$\min\{Pr(s_{in},\psi,Adv,0)\mid Adv\in Adv_M\}$，$\min\{Pr(s_{in},\psi,Adv,1)\mid Adv\in Adv_M\}$，…是一个递增的数列，且 1 为其上界，因此该序列必收敛. 收敛性保证了判断准则 1 在不预先设置计算步长 m 的情况下的终止性. 上述过程存在这样一个问题，即如何计算 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$，在 3.4 小节我们将探讨 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$ 的计算问题.

3.4　限界模型检测算法

算法的主要思想是设计一套线性方程组刻画不同算子的限界语义的算法，使得线性方程组的解即为该算子对应的概率度量，从而 PCTL 限界语义的可满足性检测可通过判断线性方程组的解是否满足相应的概率度量约束即可.

对于公式 $P_{\propto p}(\psi)$，我们主要通过计算 $p^{\min}_{s,k}(\psi)=\min\{Pr(s,\psi,Adv,k)\mid Adv\in Adv_M\}$ 来完成验证过程. 对于公式 $\phi=P^{\exists}_{\propto p}(\psi)$，主要通过计算 $p^{\max}_{s,k}(\psi)=\max\{Pr(s,\psi,Adv,k)\mid Adv\in Adv_M\}$ 来完成验证过程. 设 $p^{\min}_{s,k}(\psi)=p$，则对任意的 $0\leq r\leq p$，$s\models P_{\geq r}(\psi)$，对任意的 $0\leq l<p$，$s\models P_{>l}(\psi)$. 令 $k\geq 0$ 为限界模型检测的界，$S_{\phi,k}=\{s\in S\mid s\models_k\phi\}$. 对 $PCTL^{\exists}_{\geq}$ 公式 ϕ，引入记号 $y(s,\phi,k)\in\{0,1\}$ 来表示 $s\models_k\phi$ 是否成立：$y(s,\phi,k)=1$ 表示 $s\models_k\phi$；$y(s,\phi,k)=0$ 表示 $s\not\models_k\phi$. $y(s,\phi,k)$ 定义如下：

ϕ 是原子命题：如果 $\phi\in L(s)$，则 $y(s,\phi,k)=1$，否则 $y(s,\phi,k)=0$.

ϕ 是原子命题：如果 $\phi\in L(s)$，则 $y(s,\neg\phi,k)=1$，否则 $y(s,\neg\phi,k)=0$.

$\phi=\phi_1\vee\phi_2$：$y(s,\phi,k)=y(s,\phi_1,k)\vee y(s,\phi_2,k)$.

$\phi=\phi_1\wedge\phi_2$：$y(s,\phi,k)=y(s,\phi_1,k)\wedge y(s,\phi_2,k)$.

$\phi=P_{\propto p}(\psi)$：如果 $p^{\min}_{s,k}(\psi)\propto p$，则 $y(s,\phi,k)=1$，否则 $y(s,\phi,k)=0$.

$\phi=P^{\exists}_{\propto p}(\psi)$：如果 $p^{\max}_{s,k}(\psi)\propto p$，则 $y(s,\phi,k)=1$，否则 $y(s,\phi,k)=0$.

对于 $p^{\min}_{s,k}(\psi)$ 和 $p^{\max}_{s,k}(\psi)$，时态算子的语义不同决定了转换方法不同，下面分别讨论.

情况 1. ψ 为原子命题.

如果 $\psi\in L(s)$，则 $p^{\min}_{s,k}(\psi)=1$，否则 $p^{\min}_{s,k}(\psi)=0$.

情况 2. $\psi=X\delta$.

当 $k=0$ 时，由于当前状态 s 没有后继状态，因此 $p^{\min}_{s,0}(\psi)=0$；

当 $k\geq 1$ 时，首先需要对每一个调度 Adv 计算 $Pr(s,\psi,Adv,k)$，然后取其中的最小值，因此

$$p^{\min}_{s,k}(\psi)=\min_{(a,\mu)\in Steps(s)}\Big\{\sum_{s'\in S}y(s',\delta,k-1)\mu(s')\Big\}.$$

情况 3. $\psi=F\delta$.

当 $k=0$ 时，$p^{\min}_{s,0}(\psi)$ 完全依赖于 s 是否满足 δ，因此如果 $y(s,\delta,0)=1$，则 $p^{\min}_{s,0}(\psi)=1$，否则 $p^{\min}_{s,0}(\psi)=0$；

当 $k\geq 1$ 时，$p^{\min}_{s,k}(\psi)$ 的计算分为两部分，即当前状态 s 满足 δ 和不满足 δ. 在满足的情形下 $p^{\min}_{s,k}(\psi)=1$，否则 $p^{\min}_{s,k}(\psi)$ 由 s 的后继状态决定，因此

$$p^{\min}_{s,k}(\psi)=y(s,\delta,k)+(1-y(s,\delta,k))\cdot\min_{(a,\mu)\in Steps(s)}\Big\{\sum_{s'\in S}\mu(s')p^{\min}_{s',k-1}(\psi)\Big\}.$$

情况 4. $\psi=G\delta$.

当 $k=0$，$y(s,\delta,0)=0$ 时，s 不满足 δ，因此 $p^{\min}_{s,0}(\psi)=0$；

当 $k=0$，$y(s,\delta,0)=1$ 时，由于 G 算子要求考察无穷长的路径，因此如果存在 $(a,\mu)\in Steps(s)$ 使得 $\mu(s)<1$，则 $p^{\min}_{s,0}(\psi)=0$，否则 $p^{\min}_{s,0}(\psi)=1$；

当 $k\geq 1$ 时，采用一种下近似的计算方法，即如果在一个循环中存在概率小于 1 的状态转换，则包含此循环的路径的概率度量为 0，因此

$$p^{\min}_{s,k}(\psi)=\min_{\substack{(a_1,\mu_1)\in Steps(s_0),\cdots,\\(a_{k+1},\mu_{k+1})\in Steps(s_k)}}\sum_{i=0}^{k}\sum_{s_0,\cdots,s_k\in S}y(s_0,\delta,k)\cdot y(s_1,\delta,k)\cdot\mu_1(s_1)\cdot\cdots\cdot y(s_i,\delta,k)\cdot\mu_i(s_i)\cdot y(s_{i+1},\delta,k)\cdot\lfloor\mu_{i+1}(s_{i+1})\rfloor\cdot\cdots\cdot y(s_k,\delta,k)\cdot\lfloor\mu_k(s_k)\rfloor\cdot\lfloor\mu_{k+1}(s_i)\rfloor,$$

这里记号 $\lfloor\mu_j(s_j)\rfloor$ 表示对 $\mu_j(s_j)$ 取整 $(i+1\leq j\leq k+1)$.

情况 5. $\psi=\varphi U\gamma$.

当 $k=0$ 时，$p^{\min}_{s,0}(\psi)$ 完全依赖于 s 是否满足 γ，

因此如果 $y(s,\gamma,0)=1$，则 $p_{s,0}^{\min}(\psi)=1$，否则 $p_{s,0}^{\min}(\psi)=0$；

当 $k\geq 1$ 时，分成两种情况，(1) s 满足 γ，此时 $p_{s,k}^{\min}(\psi)=1$；(2) s 满足 φ，不满足 γ，此时

$$p_{s,k}^{\min}(\psi)=y(s,\varphi,k)\cdot\min_{(a,\mu)\in Steps(s)}\left\{\sum_{s'\in S}\mu(s')p_{s',k-1}^{\min}(\psi)\right\}.$$

情况 6. $\psi=\varphi R\gamma$.

依据 R 算子的语义可以分解成两种情形讨论，其一类似于 U 算子，其二类似于 G 算子，具体计算过程如下：

当 $k=0$ 时，$y(s,\gamma,0)=0$，则 $p_{s,0}^{\min}(\psi)=0$；

当 $k=0$ 时，如果 $y(s,\varphi,0)=y(s,\gamma,0)=1$，则 $p_{s,0}^{\min}(\psi)=1$；

当 $k=0$，$y(s,\gamma,0)=1$，$y(s,\varphi,0)=0$ 时，如果存在 $(a,\mu)\in Steps(s)$ 使得 $\mu(s)<1$，则 $p_{s,0}^{\min}(\psi)=0$，否则 $p_{s,0}^{\min}(\psi)=1$；

当 $k\geq 1$ 时，因为 $\psi=\varphi R\gamma\equiv G\gamma\vee(\gamma U(\gamma\wedge\varphi))$，故分成两部分

$$p_{s,k}^{\min}(\psi)=$$

$$\min_{\substack{(a_1,\mu_1)\in Steps(s_0),\cdots,\\(a_{k+1},\mu_{k+1})\in Steps(s_k)}}\sum_{i=0}^{k}\sum_{s_0,\cdots,s_k\in S}(1-y(s_0,\varphi,k))\cdot\cdots\cdot(1-y(s_k,\varphi,k))\cdot y(s_0,\gamma,k)\cdot y(s_1,\gamma,k)\cdot\mu_1(s_1)\cdot\cdots\cdot y(s_i,\gamma,k)\cdot\mu_i(s_i)\cdot y(s_{i+1},\gamma,k)\cdot\lfloor\mu_{i+1}(s_{i+1})\rfloor\cdot\cdots\cdot y(s_k,\delta,k)\cdot\lfloor\mu_k(s_k)\rfloor\cdot\lfloor\mu_{k+1}(s_i)\rfloor+p_{s,k}^{\min}(\gamma U(\gamma\wedge\varphi)),$$

这里加入因子 $(1-y(s_j,\varphi,k))(0\leq j\leq k)$ 的主要目的是避免重复计算 $\{\pi\mid\pi\models_k G\gamma\wedge\pi\models_k\gamma U(\gamma\wedge\varphi)\}$ 的概率度量.

对于公式 $\psi=P_{\infty p}^{\exists}(\psi)$ 具体的计算过程如下.

情况 1. ψ 为原子命题.

如果 $\psi\in L(s)$，则 $p_{s,k}^{\max}(\psi)=1$，否则 $p_{s,k}^{\max}(\psi)=0$.

情况 2. $\psi=X\delta$.

当 $k=0$ 时，由于当前状态 s 没有后继状态，因此 $p_{s,0}^{\max}(\psi)=0$；

当 $k\geq 1$ 时，首先需要对每一个调度 Adv 计算 $Pr(s,\psi,Adv,k)$，然后取其中的最大值，因此

$$p_{s,k}^{\max}(\psi)=\max_{(a,\mu)\in Steps(s)}\left\{\sum_{s'\in S}y(s',\delta,k-1)\mu(s')\right\}.$$

情况 3. $\psi=F\delta$.

当 $k=0$ 时，$p_{s,0}^{\max}(\psi)$ 完全依赖于 s 是否满足 δ，因此如果 $y(s,\delta,0)=1$，则 $p_{s,0}^{\max}(\psi)=1$，否则 $p_{s,0}^{\max}(\psi)=0$；

当 $k\geq 1$ 时，$p_{s,k}^{\max}(\psi)$ 的计算分为两部分，即当前状态 s 满足 δ 和不满足 δ. 在满足的情形下之 $p_{s,k}^{\max}(\psi)=1$，否则 $p_{s,k}^{\max}(\psi)$ 由 s 的后继状态决定，因此

$$p_{s,k}^{\max}(\psi)=y(s,\delta,k)+(1-y(s,\delta,k))\max_{(a,\mu)\in Steps(s)}\left\{\sum_{s'\in S}\mu(s')p_{s',k-1}^{\max}(\psi)\right\}.$$

情况 4. $\psi=G\delta$.

当 $k=0$，$y(s,\delta,0)=0$ 时，s 不满足 δ，因此 $p_{s,0}^{\max}(\psi)=0$；

当 $k=0$，$y(s,\delta,0)=1$ 时，由于 G 算子要求考察无穷长的路径，因此如果存在 $(a,\mu)\in Steps(s)$ 使得 $\mu(s)=1$，则 $p_{s,0}^{\max}(\psi)=1$，否则 $p_{s,0}^{\max}(\psi)=0$；

当 $k\geq 1$ 时，采用一种下近似的计算方法，即如果在一个循环中存在概率小于 1 的状态转换，则包含此循环的路径的概率度量为 0，因此

$$p_{s,k}^{\max}(\psi)=$$

$$\max_{\substack{(a_1,\mu_1)\in Steps(s_0),\cdots,\\(a_{k+1},\mu_{k+1})\in Steps(s_k)}}\sum_{i=0}^{k}\sum_{s_0,\cdots,s_k\in S}y(s_0,\delta,k)\cdot y(s_1,\delta,k)\cdot\mu_1(s_1)\cdot\cdots\cdot y(s_i,\delta,k)\cdot\mu_i(s_i)\cdot y(s_{i+1},\delta,k)\cdot\lfloor\mu_{i+1}(s_{i+1})\rfloor\cdot\cdots\cdot y(s_k,\delta,k)\cdot\lfloor\mu_k(s_k)\rfloor\cdot\lfloor\mu_{k+1}(s_i)\rfloor,$$

这里记号 $\lfloor\mu_j(s_j)\rfloor$ 表示对 $\mu_j(s_j)$ 取整 $(i+1\leq j\leq k+1)$.

情况 5. $\psi=\varphi U\gamma$.

当 $k=0$ 时，$p_{s,0}^{\max}(\psi)$ 完全依赖于 s 是否满足 γ，因此如果 $y(s,\gamma,0)=1$，则 $p_{s,0}^{\max}(\psi)=1$，否则 $p_{s,0}^{\max}(\psi)=0$；

当 $k\geq 1$ 时，分成两种情况，(1) s 满足 γ，此时 $p_{s,k}^{\max}(\psi)=1$；(2) s 满足 φ，不满足 γ，此时

$$p_{s,k}^{\max}(\psi)=y(s,\varphi,k)\cdot\max_{(a,\mu)\in Steps(s)}\left\{\sum_{s'\in S}\mu(s')p_{s',k-1}^{\max}(\psi)\right\};$$

情况 6. $\psi=\varphi R\gamma$.

依据 R 算子的语义可以分解成两种情形讨论，其一类似于 U 算子，其二类似于 G 算子，具体计算过程如下：

当 $k=0$ 时，$y(s,\gamma,0)=0$，则 $p_{s,0}^{\max}(\psi)=0$；

当 $k=0$ 时，如果 $y(s,\varphi,0)=y(s,\gamma,0)=1$，则 $p_{s,0}^{\max}(\psi)=1$；

当 $k=0$，$y(s,\gamma,0)=1$，$y(s,\varphi,0)=0$ 时，如果存在 $(a,\mu)\in Steps(s)$ 使得 $\mu(s)=1$，则 $p_{s,0}^{\max}(\psi)=1$，否则 $p_{s,0}^{\min}(\psi)=0$；

当 $k\geq 1$ 时，因为 $\psi=\varphi R\gamma\equiv G\gamma\vee(\gamma U(\gamma\wedge\varphi))$，故分成两部分

$$p_{s,k}^{\max}(\psi)=$$

$$\max_{\substack{(a_1,\mu_1)\in Steps(s_0),\cdots,\\(a_{k+1},\mu_{k+1})\in Steps(s_k)}}\sum_{i=0}^{k}\sum_{s_0,\cdots,s_k\in S}(1-y(s_0,\varphi,k))\cdot\cdots\cdot(1-y(s_k,\varphi,k))\cdot y(s_0,\gamma,k)\cdot y(s_1,\gamma,k)\cdot\mu_1(s_1)\cdot\cdots\cdot y(s_i,\gamma,k)\cdot\mu_i(s_i)\cdot y(s_{i+1},\gamma,k)\cdot\lfloor\mu_{i+1}(s_{i+1})\rfloor\cdot\cdots\cdot$$

$y(s_k,\delta,k)\cdot\lfloor\mu_k(s_k)\rfloor\cdot\lfloor\mu_{k+1}(s_i)\rfloor+p^{\min}_{s,k}(\gamma\mathrm{U}(\gamma\wedge\varphi))$.

现在分析线性方程组的阶与 MDP 的大小、界、公式长度等元素相互之间的依赖关系.

定义 9.

0 步可达：称 s 是从自身出发 0 步可达的；

1 步可达：如果存在 $(a,\mu)\in Steps(s)$ 使得 $\mu(s_1)>0$，则称 s_1 是从 s 出发 1 步可达的；

l 步可达：如果 s_{l-1} 是从 s 出发 $l-1$ 步可达的，且存在 $(a,\mu)\in Steps(s_{l-1})$ 使得 $\mu(s_l)>0$，则称 s_l 是从 s 出发 l 步可达.

对 PCTL$^{\exists}_{\geq}$ 公式 ϕ，令 $|\phi|$ 表示 ϕ 中出现的符号的数目. 设 $M=(S,s_{\mathrm{in}},Act,Steps,Ap,L)$ 为 MDP，ϕ 是被分析的公式，k 为界，所有从初始状态 i 步可达的状态数目记为 N_i，线性方程组中变元的数量记为 V. 在不同状态下，对 ϕ 的每一个子公式转换算法建立了其与每一个不大于 k 的界的组合. 此外每个 ϕ 的子公式 φ 与变元 $y(s,\varphi,k)$ 建立了一一对应关系. 上述分析过程表明 $V\leq(N_0+\cdots+N_k)\times|\phi|\times k\times 2$.

直接法和迭代法是求解线性方程组的两类主要方法，直接法的特点是准确性和可靠性高，迭代法的特点是适用于高阶的方程组. 而对于 MDP，转换算法演绎出的是上三角方程组. 因此可忽略方程组阶的影响，仍然选择直接法来求解方程. 变元求解的次序可采用文献[23]中定义的语法树来确定，即先求解原子命题对应的变元，然后由里向外以此求解各子公式对应的变元.

4　实例研究

在分布式系统中当进程或者用户进入非法状态时，我们总希望其能尽快回归到合法状态. 为此 Israeli 和 Jalfon 提出自稳定协议[25]，从而保证当系统进入非法状态时，在没有外部力量的帮助下能够在有限步内自动回到合法状态，这里我们将合法状态称为稳定状态. 假设系统由 N 个独立的进程 P_1，P_2，…，P_N 组成，进程之间的运行是异步的. 稳定状态指的是只有一个进程享有特权，这里享有特权是指该进程拥有一个 token.

每个进程 P_i 使用一个布尔变量 q_i 说明该进程是否拥有一个 token. 当进程拥有 token 时称该进程是活的. 只有活的进程才能被调度. 进程被调度以后，该进程随机的将 token 移给到它左边的或者右边的进程. 如果进程拥有的 token 数超过一个，将被合并为一个.

现在探讨如何利用马尔可夫决策过程模拟该协议. 具体决策过程 $M=(S,s_{\mathrm{in}},Act,Steps,Ap,L)$ 定义如下：

$S=B^N$，这里 $B=\{q_1,q_2,\cdots,q_N\}$，即每个状态表示为一个布尔向量 $\boldsymbol{s}=(q_1,q_2,\cdots,q_N)$，其中第 i 个元素表示进程 P_i 是否拥有 token；

$s_{\mathrm{in}}=\{1,\cdots,1\}$ 是初始状态，即每个进程均拥有一个 token；

$Act=\{1,2,\cdots,N\}$ 是动作集，这里 i 表示进程 P_i 被调度；

$Steps:S\to 2^{Act\times Dist(S)}$，对任意的状态 $\boldsymbol{s}\in S$，$(i,\mu)\in Steps(\boldsymbol{s})$ 当且仅当 $q_i=1$，任意的 $\boldsymbol{s}'\in S$ 满足 $\mu(s')=$

$$\begin{cases}\frac{1}{2},\text{ if } q'_{i\oplus 1}=1,q'_i=0,\text{ and } q'_j=q_j \text{ for all } j\neq i,i\oplus 1\\ \frac{1}{2},\text{ if } q'_{i\odot 1}=1,q'_i=0,\text{ and } q'_j=q_j \text{ for all } j\neq i,i\odot 1\\ 0,\ \text{其它}\end{cases}$$

这里 $i\oplus 1=i+1(\mathrm{mod}\ N)$，$i\odot 1=\begin{cases}N, & \text{if } i=1\\ i-1, & \text{其它}\end{cases}$；

$Ap=\{q_1,q_2,\cdots,q_N,stable\}$；

$L:S\to 2^{Ap}$，这里 $q_i\in L(\boldsymbol{s})$ 当且仅当 $q_i=1$，$stable\in L(\boldsymbol{s})$ 当且仅当 $\exists 1\leq i\leq N$ 使得 $q_i=1$ 且 $\forall 1\leq j\leq N(j\neq i\to q_j=0)$.

我们验证这样的属性：令 L 表示一步可达状态中稳定状态从其可达的概率不低于 0.5 的所有状态，则 L 从初始状态可达的概率也不低于 0.5. 该属性利用 PCTL 逻辑描述为 $P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$. 我们检测当界为 2 时，$s_0$ 是否满足 $P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$，为此需要计算概率度量 $p^{\min}_{s_0,2}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$，即 $\min\{Pr(s_0,\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable),Adv,2)\mid Adv\in Adv_M\}$. 界为 2 时自稳定协议对应的马尔可夫决策过程展开图如图 2 所示. 具体的计算过程如下，主要是按照语法树由上向下逐层推进，直到界为 0 或者原子命题为止.

$$(1)\ p^{\min}_{s_0,2}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))=\min\Big\{\frac{1}{2}y(s_1,P_{\geq\frac{1}{2}}(\mathrm{F}stable),1)+\frac{1}{2}y(s_1,P_{\geq\frac{1}{2}}(\mathrm{F}\,stable),1),\frac{1}{2}y(s_2,P_{\geq\frac{1}{2}}(\mathrm{F}stable),1)+\frac{1}{2}y(s_2,P_{\geq\frac{1}{2}}(\mathrm{F}stable),1),\frac{1}{2}y(s_3,P_{\geq\frac{1}{2}}(\mathrm{F}\,stable),1)+\frac{1}{2}y(s_3,P_{\geq\frac{1}{2}}(\mathrm{F}stable),$$

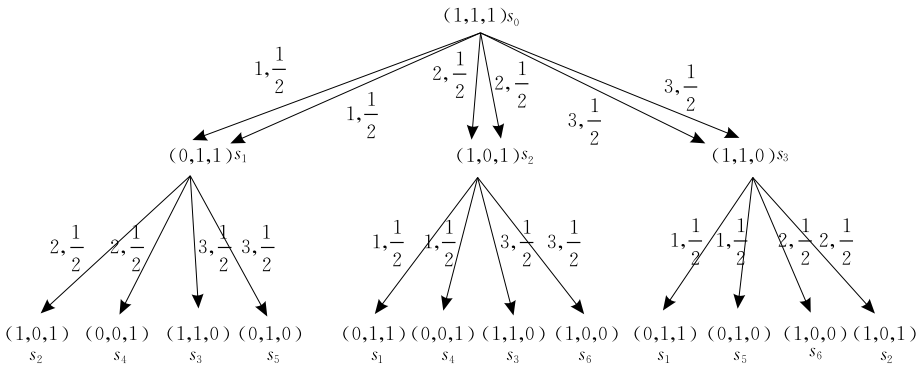

图 2 界为 2 时自稳定协议对应的 MDP 展开图

$1)\Big\} = \min\{y(s_1, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) + y(s_2, P_{\geq\frac{1}{2}}(\mathrm{F}stable), 1) + y(s_3, P_{\geq\frac{1}{2}}(\mathrm{F}stable), 1)\}$.

(2) $p^{\min}_{s_1,1}(\mathrm{F}\ stable) = y(s_1, stable, 0) + (1 - y(s_1, stable, 0)) \times \min\Big\{\frac{1}{2}p^{\min}_{s_2,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_4,0}(\mathrm{F}\ stable), \frac{1}{2}p^{\min}_{s_3,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_5,0}(\mathrm{F}\ stable)\Big\}$.

(3) 如果 $p^{\min}_{s_1,1}(\mathrm{F}\ stable) \geq \frac{1}{2}$,则 $y(s_1, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 1$;否则 $y(s_1, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 0$.

(4) $p^{\min}_{s_2,1}(\mathrm{F}\ stable) = y(s_2, stable, 0) + (1 - y(s_2, stable, 0)) \times \min\Big\{\frac{1}{2}p^{\min}_{s_1,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_4,0}(\mathrm{F}\ stable), \frac{1}{2}p^{\min}_{s_3,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_6,0}(\mathrm{F}\ stable)\Big\}$.

(5) 如果 $p^{\min}_{s_2,1}(\mathrm{F}\ stable) \geq \frac{1}{2}$,则 $y(s_2, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 1$;否则 $y(s_2, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 0$.

(6) $p^{\min}_{s_3,1}(\mathrm{F}\ stable) = y(s_3, stable, 0) + (1 - y(s_3, stable, 0)) \times \min\Big\{\frac{1}{2}p^{\min}_{s_1,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_5,0}(\mathrm{F}\ stable), \frac{1}{2}p^{\min}_{s_6,0}(\mathrm{F}\ stable) + \frac{1}{2}p^{\min}_{s_2,0}(\mathrm{F}\ stable)\Big\}$.

(7) 如果 $p^{\min}_{s_3,1}(\mathrm{F}\ stable) \geq \frac{1}{2}$,则 $y(s_3, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 1$;否则 $y(s_3, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1) = 0$.

(8) $y(s_1, stable, 0) = y(s_2, stable, 0) = y(s_3, stable, 0) = 0$; $y(s_4, stable, 0) = y(s_5, stable, 0) = y(s_6, stable, 0) = 1$.

(9) $p^{\min}_{s_1,0}(\mathrm{F}\ stable) = p^{\min}_{s_2,0}(\mathrm{F}\ stable) = p^{\min}_{s_3,0}(\mathrm{F}\ stable) = 0$; $p^{\min}_{s_4,0}(\mathrm{F}\ stable) = p^{\min}_{s_5,0}(\mathrm{F}\ stable) = p^{\min}_{s_6,0}(\mathrm{F}\ stable) = 1$.

对于上述方程组,变元求解的顺序为

(1) $y(s_1, stable, 0), y(s_2, stable, 0), y(s_3, stable, 0), y(s_4, stable, 0), y(s_5, stable, 0), y(s_6, stable, 0)$.

(2) $p^{\min}_{s_2,0}(\mathrm{F}\ stable), p^{\min}_{s_3,0}(\mathrm{F}\ stable), p^{\min}_{s_1,0}(\mathrm{F}\ stable), p^{\min}_{s_6,0}(\mathrm{F}\ stable), p^{\min}_{s_4,0}(\mathrm{F}\ stable), p^{\min}_{s_5,0}(\mathrm{F}\ stable)$.

(3) $p^{\min}_{s_1,1}(\mathrm{F}\ stable), p^{\min}_{s_2,1}(\mathrm{F}\ stable), p^{\min}_{s_3,1}(\mathrm{F}\ stable)$.

(4) $y(s_1, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1), y(s_2, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1), y(s_3, P_{\geq\frac{1}{2}}(\mathrm{F}\ stable), 1)$.

(5) $p^{\min}_{s_0,2}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\ stable))$.

最终得出 $p^{\min}_{s_0,2}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\ stable)) = \frac{1}{2}$,所以 $s_0 \models P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\ stable))$.

5 实验结果

为了考察限界检测技术在实际应用中约简状态空间的效果,利用限界模型检测算法验证了 3 个实例:(1) 第 4 部分的自稳定协议;(2) Lehmann 和 Robin 提出的解决哲学家就餐问题的策略;(3) 带冲突避免的载波监听多路访问协议 CSMA/CA. 表 2、表 3 和表 4 分别给出了线性方程组变元数目随着进程数(主体数)、界的变化而变化的情况. 表中的变元数目是依据 3.4 节中描述的变元数目与马尔可夫决策过程的状态展开空间、公式长度以及界之间的关系估算出的上界.

表中的几个属性说明如下:

$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\ stable))$:令事件 E 表示一步可达状态到达稳定状态的概率不低于 0.5,则事件 E 成立的概率不低于 0.5.

$hungry \rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$:哲学家饥饿后最终吃到晚餐的概率不低于 0.5;

$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\ eat))$:令事件 E 表示任何时候哲学家想吃晚餐最终都能吃到晚餐的概率不低于 0.8,则事件 E 成立的概率不低于 0.75;

$P_{\geq 1}(True\,\mathrm{U}(s1 = 12 \wedge s2 = 12))$:数据成功发送的概率为 1.

表 2　自稳定协议的限界模型检测与无界模型检测比较

协议	属性	进程	界	变元	初始状态	全局状态	转换关系
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	3	2	<56	1	7	21
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	3	3	<68	1	7	21
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	4	2	<86	1	15	56
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	4	3	<170	1	15	56
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	5	3	<272	1	31	140
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	5	4	<452	1	31	140
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	6	3	<410	1	63	336
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	6	4	<746	1	63	336
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	7	4	<1178	1	127	784
self-stabilising protocol	$P_{\geq\frac{1}{2}}(\mathrm{X}P_{\geq\frac{1}{2}}(\mathrm{F}\,stable))$	7	5	<1892	1	127	784

表 3　哲学家就餐问题的限界模型检测与无界模型检测比较

协议	属性	主体个数	界	变元	初始状态	全局状态	转换关系
Randomized dining philosophers	$hungry\rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$	3	3	<470	1	956	3048
Randomized dining philosophers	$hungry\rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$	4	3	<1010	1	9440	40120
Randomized dining philosophers	$hungry\rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$	5	3	<1862	1	93068	49420
Randomized dining philosophers	$hungry\rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$	6	3	<3098	1	917424	5848524
Randomized dining philosophers	$hungry\rightarrow P_{\geq\frac{1}{2}}((True)\mathrm{U}(eat))$	7	3	<4790	1	9043420	67259808
Randomized dining philosophers	$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\,eat))$	3	3	<940	1	956	3048
Randomized dining philosophers	$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\,eat))$	4	3	<2020	1	9440	40120
Randomized dining philosophers	$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\,eat))$	5	3	<3724	1	93068	49420
Randomized dining philosophers	$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\,eat))$	6	3	<6196	1	917424	5848524
Randomized dining philosophers	$P_{\geq\frac{3}{4}}(\mathrm{G}P_{\geq\frac{4}{5}}(\mathrm{F}\,eat))$	7	3	<9580	1	9043420	67259808

表 4　CSMA/CA 协议的限界模型检测与无界模型检测比较

协议	属性	进程	界	最大退避次数	变元	初始状态	全局状态
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	0	<108	1	16069
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	1	<108	1	34855
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	2	<108	1	87345
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	3	<108	1	217082
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	4	<108	1	586255
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	5	<108	1	1774068
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	3	6	<108	1	5958233
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	0	<144	1	16069
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	1	<144	1	34855
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	2	<144	1	87345
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	3	<144	1	217082
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	4	<144	1	586255
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	5	<144	1	1774068
CSMA/CA	$P_{\geq1}(True\,\mathrm{U}(s1=12\wedge s2=12))$	5	4	6	<144	1	5958233

全局检测算法与限界检测算法状态空间可以显式表示也可以隐式表示，如符号化技术. 因此我们避开状态空间表示的具体数据结构，从所需遍历的状态空间规模的角度比较限界检测与全局检测算法对空间的需求. 为此表 2～表 4 同时给出了利用模型检测工具 PRISM[24] 计算出的在全局状态空间下每个实例的初始状态数目，状态的数目以及转换关系的数目.

从表 2～表 4 可以看出马尔可夫限界模型检测具有以下几个方面的特点：

(1)该技术是一种前向搜索技术，不需要访问所有的空间，且可以快速发现属性成立的证据.

(2)在证据较短的情况下，所需内存空间少于基于 MTBDD 的符号化模型检测技术.

(3)与基于 MTBDD 的方法不一样，该技术不需要对变量进行排序.

6 终止标准的修正

理论上，$s \models_k \phi \Rightarrow s \models_{k+1} \phi$，因此随着界的增长概率度量会逐渐递增. 我们首先以自稳定协议和图 3 所示的马尔可夫决策过程 M_3 为例，研究随着界的增长概率度量增长的规律. 在 M_3 中验证的属性为 $P_{\geq \frac{5}{9}}(\mathrm{F}r)$，这里 $r \in L(s_2) \cap L(s_5) \cap L(s_8)$，$r \notin L(s_0) \cup L(s_1) \cup L(s_3) \cup L(s_4) \cup L(s_6) \cup L(s_7) \cup L(s_8)$. 图 4 中实线给出了在主体为 3 的情况下，概率度量 $p^{\min}_{s_0,k}(\mathrm{X}P_{\geq \frac{1}{2}}(\mathrm{F}\ stable))$ 的递增规律，虚线给出了概率度量 $p^{\min}_{s_0,k}(\mathrm{F}r)$ 的递增规律. 图 4 中的两种曲线分别代表了概率度量增长的典型规律. 令 p_i 表示界为 i 时得到的概率度量. 第 1 种规律是 $p_0, p_1, \cdots$ 是从某点开始的严格递增序列，即存在 $j \geq 0$ 使得对任意的 $i \geq j$，$p_i < p_{i+1}$，第 2 种规律是 $p_0, p_1, \cdots$ 为非严格递增序列，即对任意的 $j \geq 0$，存在 $i \geq j$ 使得 $p_i = p_{i+1}$.

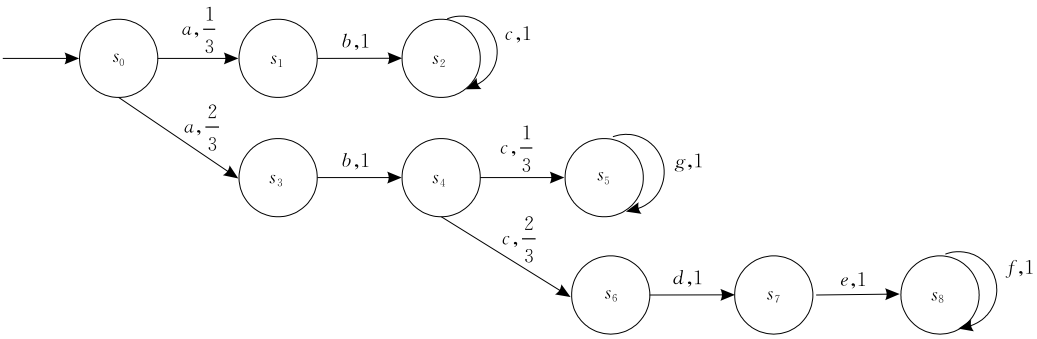

图 3 MDP M_3

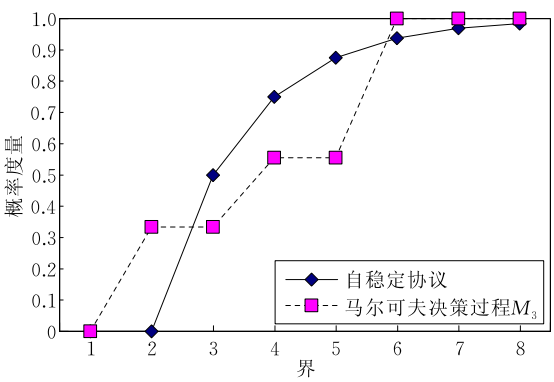

图 4 概率度量增长的规律

终止判断准则 1 的核心是相连两次概率度量的差控制在预先设置的范围内时计算过程结束. 对于数值序列 $x_0, x_1, \cdots$，当对任意的自然数 i，$|x_{i+2} - x_{i+1}| < |x_{i+1} - x_i|$ 时准则 1 是有效的. 但是对于表 4 中的两个概率度量序列都无效. 对自稳定协议，当界为 1 时检测过程终止，此时概率度量为 0；对于 M_3，当界为 3 时终止，此时概率度量为 1/3. 对上述两个实例，按照准则 1 得到的近似概率度量与真实的概率度量误差较大，因此必须对终止标准进行修正，从而使终止准则适用于图 4 中的曲线.

修正方案 1. 比较界连续的多次概率度量之和. 设 n, k 为自然数，在第 k 步如果

$$\sum_{j=k-n+1}^{k} \min\{Pr(s_{\mathrm{in}}, \psi, Adv, j) \mid Adv \in Adv_M\} - \sum_{j=k-2n+1}^{k-n} \min\{Pr(s_{\mathrm{in}}, \psi, Adv, j) \mid Adv \in Adv_M\} < \xi,$$

则检测过程终止. 具体过程如判断准则 2 所示.

判断准则 2. $\mathrm{PCTL}^{\exists}_{\geq}$ 限界模型检测的终止性判断(以修正方案 1 为终止标准).

输入：MDP $M=(S, s_{\mathrm{in}}, Act, Steps, Ap, L)$，$\mathrm{PCTL}^{\exists}_{\geq}$ 路径公式 ψ，预先设置的终止标准 ξ，预先设置的计算步长 m，自然数 n

输出：$\min\{Pr(s_{\mathrm{in}}, \psi, Adv) \mid Adv \in Adv_M\}$ 的近似值

1. 计算 $\min\{Pr(s_{\mathrm{in}}, \psi, Adv, 0) \mid Adv \in Adv_M\}$，$\min\{Pr(s_{\mathrm{in}}, \psi, Adv, 1) \mid Adv \in Adv_M\}, \cdots, \min\{Pr(s_{\mathrm{in}}, \psi, Adv, n) \mid Adv \in Adv_M\}$；

2. 令 $k=2n-1$

3. While $\sum_{j=k-n+1}^{k} \min\{Pr(s_{\mathrm{in}}, \psi, Adv, j) \mid Adv \in Adv_M\} - \sum_{j=k-2n+1}^{k-n} \min\{Pr(s_{\mathrm{in}}, \psi, Adv, j) \mid Adv \in Adv_M\} \geq \xi \wedge k \leq m$ do

{令 $k=k+1$，计算 $\min\{Pr(s_{\mathrm{in}}, \psi, Adv, k) \mid Adv \in Adv_M\}$；}

4. 输出 $\min\{Pr(s_{\mathrm{in}}, \psi, Adv, k) \mid Adv \in Adv_M\}$.

在判断准则 2 中取 $n=2$ 可避免两个测试用例中的收敛问题.限界检测计算的概率度量序列是非严格递增的,因此 n 的值越大得出的度量越逼近真实值.这种方案的主要缺点在于需要预先设定 n 的值,而且 n 的最佳取值无法确定.

修正方案 2. 比较概率度量和的比值.

设 k 为自然数,在第 k 步如果

$$\frac{\sum_{j=k-n+1}^{k}\min\{Pr(s_{in},\psi,Adv,j)\mid Adv\in Adv_M\}}{\sum_{j=k-2n+1}^{k-n}\min\{Pr(s_{in},\psi,Adv,j)\mid Adv\in Adv_M\}}-1<\xi,$$

则检测过程终止.具体过程如判断准则 3 所示.

判断准则 3. $\mathrm{PCTL}_{\geq}^{\exists}$ 限界模型检测(以修正方案 2 为终止标准).

输入：MDP $M=(S,s_{in},Act,Steps,Ap,L)$,$\mathrm{PCTL}_{\geq}^{\exists}$ 路径公式 ψ,预先设置的终止标准 ξ,预先设置的计算步长 m,自然数 n

输出：$\min\{Pr(s_{in},\psi,Adv)\mid Adv\in Adv_M\}$ 的近似值

1. 计算 $\min\{Pr(s_{in},\psi,Adv,0)\mid Adv\in Adv_M\}$,
   $\min\{Pr(s_{in},\psi,Adv,1)\mid Adv\in Adv_M\}$,
   $\min\{Pr(s_{in},\psi,Adv,2)\mid Adv\in Adv_M\}$;

2. 令 $k=2n-1$

3. While $\dfrac{\sum_{j=k-n+1}^{k}\min\{Pr(s_{in},\psi,Adv,j)\mid Adv\in Adv_M\}}{\sum_{j=k-2n+1}^{k-n}\min\{Pr(s_{in},\psi,Adv,j)\mid Adv\in Adv_M\}}-1\geq\xi\wedge k\leq m$ do

{令 $k=k+1$,计算 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$;}

4. 输出 $\min\{Pr(s_{in},\psi,Adv,k)\mid Adv\in Adv_M\}$.

判断准则 3 亦避免了两个测试用例中的收敛问题,但是 n 的取值问题与判断准则 2 一样.

7　结　论

为了克服模型检测 MDP 中的状态空间爆炸问题,本文提出了在 MDP 上 PCTL 的限界检测技术.在具有马尔可夫性的随机系统模型中,MDP 的主要特性在于具有非确定选择描述能力,在具体工作上本文结合该特性分别研究了概率计算树逻辑的限界语义、基于概率度量序列演化规律的检测过程终止判断准则、基于线性方程组求解的限界检测算法.进一步通过实验,说明了限界模型检测在属性为真的证据比较短的情况下,能快速验证属性,而且需求的空间比无界模型检测技术少.未来的主要工作是实现限界模型检测算法的符号化执行过程,同时挖掘 MDP 的结构,待验证的属性等因素与终止标准的关系,为设置一个完备的终止标准奠定基础.

参　考　文　献

[1] Clarke E M, Grumberg O, Peled D. Model Checking. MA: MIT Press, 1999

[2] Lin Hui-Min, Zhang Wen-Hui. Model checking: Theories, techniques and applications. Acta Electronica Sinica, 2002, 30(12A): 9-14(in Chinese)
(林惠民，张文辉. 模型检测：理论、方法与应用. 电子学报, 2002, 30(12A): 9-14)

[3] Hansson H, Jonsson B. A logic for reasoning about time and reliability. Formal Aspects of Computing, 1994, 6(5): 512-535

[4] Baier C, Katoen J P. Principles of Model Checking. MA: MIT Press, 2008

[5] Rutten J, Kwiatkowska M, Norman G, Parker D. Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. Providence, RI: American Mathematical Society, 2004

[6] Clarke E M. My 27-year quest to overcome the state explosion problem//Lecture Notes in Computer Science 5330. Springer, 2008: 182-182

[7] Clarke E M, Grumberg O, Jha S, Lu Y, Veith H. Progress on the state explosion problem in model checking//Lecture Notes in Computer Science 2000. Springer, 2001: 176-194

[8] Bryant R E. Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers, 1986, 35(8): 687-691

[9] Burch J R, Clarke E M, McMillan K L. Symbolic model checking: 10^{20} states and beyond. Information and Computation, 1992, 98(2): 142-170

[10] Qu Wan-Xia, Li Tun, Guo Yang, Yang Xiao-Dong. Advances in predicate abstraction. Journal of Software, 2008, 19(1): 27-38(in Chinese)
(屈婉霞，李暾，郭阳，杨晓东. 谓词抽象技术研究. 软件学报, 2008, 19(1): 27-38)

[11] Wolper P, Godefroid P. Partial order methods for temporal verification//Lecture Notes in Computer Science 715. Springer, 1993: 233-246

[12] Emerson E A, Sistla A P. Symmetry and model checking. Formal Methods in System Design, 1996, 9(1): 105-131

[13] Pasareanu C S, Dwyer M B, Huth M. Assume-guarantee model checking of software: A comparative case study. Lecture Notes in Computer Science 1680. Springer, 1999: 168-183

[14] Biere A, Cimatti A, Clarke E M, Zhu Y. Symbolic model checking without BDDs//Lecture Notes in Computer Science 1579. Springer, 1999: 193-207

[15] Yang Jin-Ji, Su Kai-Le, Luo Xiang-Yu, Lin Han, Xiao Yin-Yin. Optimization of bounded model checking. Journal of Software, 2009, 20(8): 2005-2014(in Chinese)

(杨晋吉，苏开乐，骆翔宇，林瀚，肖茵茵. 有界模型检测的优化. 软件学报，2009，20(8)：2005-2014)

[16] Luo Xiang-Yu, Su Kai-Le, Yang Jin-Ji. Bounded model checking for temporal epistemic logic in synchronous multi-agent systems. Journal of Software, 2006, 17(12): 2585-2498(in Chinese)
(骆翔宇，苏开乐，杨晋吉. 有界模型检测同步多智体系统的时态认知逻辑. 软件学报，2006，17(12)：2585-2498)

[17] Zhang Wen-Hui. Model checking with SAT-based characterization of ACTL formulas//Lecture Notes in Computer Science 4789. Springer, 2007: 191-211

[18] Chen Wei, Zhang Wen-Hui. Bounded model checking of ACTL formulae//Proceedings of the 3rd IEEE International Symposium on Theoretical Aspects of Software Engineering. Tianjin, China, 2009. Washington D C: IEEE Computer Society Press, 2009: 90-99

[19] Xu Liang, Chen Wei, Xu Yan-Yan, Zhang Wen-Hui. Improved bounded model checking for universal fragment of CTL. Journal of Computer Science and Technology, 2009, 24(1): 96-109

[20] Penczek W, Wozna B, Zbrzezny A. Bounded model checking for the universal fragment of CTL. Fundamenta Informaticae, 2002, 51(1-2): 135-156

[21] Penczek W, Lomuscio A. Verifying epistemic properties of multi-Agent systems via bounded model checking. Fundamenta Informaticae, 2003, 55(2): 167-185

[22] Lomuscio A, Penczek W, Wozna B. Bounded model checking for knowledge and real time. Artificial Intelligence, 2007, 171(16-17): 1011-1038

[23] Zhou Cong-Hua, Liu Zhi-Feng, Wang Chang-Da. Bounded model checking for probabilistic computation tree logic. Journal of Software, 2012, 23(7): 1656-1668(in Chinese)
(周从华，刘志锋，王昌达. 概率计算树逻辑的限界模型检测. 软件学报，2012，23(7)：1656-1668)

[24] Zhou Cong-Hua, Ye Meng, Wang Chang-Da, Liu Zhi-Feng. Bounded model checking algorithm to reduce the state space in multi-agent systems. Journal of Software, 2012, 23(11): 2835-2861(in Chinese)
(周从华，叶萌，王昌达，刘志锋. 多智体系统中约简状态空间的限界模型检测算法. 软件学报，2012，23(11)：2835-2861)

[25] Israeli A, Jalfon M. Token management schemes and random walks yield self-stabilizing mutual exclusion//Proceedings of the 9th Annual ACM Symposium on Principles of Distributed Computing. Quebec, Canada, 1990. New Work: ACM Press, 1990: 119-131

[26] Kwiatkowska M, Norman G, Parker D. Probabilistic symbolic model checking with PRISM: A hybrid approach. International Journal on Software Tools for Technology Transfer, 2004, 6(2): 128-142

ZHOU Cong-Hua, born in 1978, Ph. D., associate professor. His research interests include model checking, access control, modal logic.

XING Zhi-Hu, born in 1988, M. S. candidate. His research interest is model checking.

LIU Zhi-Feng, born in 1981, Ph. D. His research interests include formal method, model checking.

WANG Chang-Da, born in 1971, Ph. D., professor. His research interest is information security techniques.

Background

The probabilistic model checking is a highly automated formal technique, which explores the full state space to complete the property analysis of the system. Currently, probabilistic model checking has been successfully used in the formal analysis of communication protocols. The state explosion problem is the key obstacle to make the probabilistic model checking feasible. In this paper, to overcome the state explosion problem, a bounded model checking technique for Markov decision processes is proposed. This work is supported by the National Natural Science Foundation of China (Nos. 61003288, 6111130184), the Ph. D. Programs Foundation of Ministry of Education of China (No. 20093227110005), and the Natural Science Foundation of Jiangsu Province (No. BK2010192).

These projects aim to provide some methods to reduce the state space when model checking is used to verify some critical systems. Our group has been working on overcoming the state explosion problem in model checking and proposed many efficient techniques such as abstraction, composition et al. And many good papers have been published in international conferences and journals such as Chinese Science F: Information Science, TAMC.